One Voice is All You Need: A One-Shot Approach to Recognize You

by

Shahriar Rumi Dipto
20141036
Priata Nowshin
20141035
Intesur Ahmed
18101685
Deboraj Chowdhury
18101242
Galib Abdun Noor
20141037

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

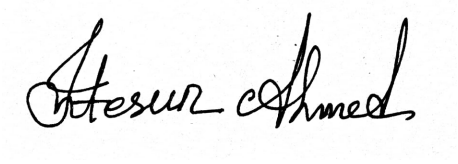
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

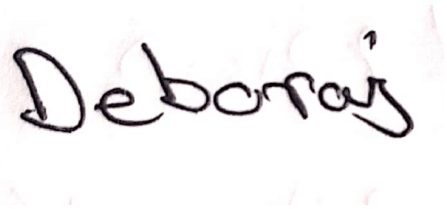
Student's Full Name & Signature:

Shahriar Rumi Dipto
20141036

Priata Nowshin
20141035



Intesur Ahmed
18101685



Deboraj Chowdhury
18101242

Galib Abdun Noor
20141037

Approval

The thesis titled "One Voice is All You Need: A One-Shot Approach to Recognize You" submitted by

1. Shahriar Rumi Dipto(20141036)
2. Priata Nowshin(20141035)
3. Intesur Ahmed(18101685)
4. Deboraj Chowdhury(18101242)
5. Galib Abdun Noor(20141037)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Human knowledge can quickly learn any unfamiliar concepts based on what they have previously learned. Keeping this in mind, researchers tested training models with limited training data in machine learning classification functions.One-shot learning has proven to be effective in the researches of Computer Vision sector, as it works accurately with a single labeled training example and a small number of training sets. By using a single input example from each class, one-shot learning can work more efficiently and quickly. For training the architecture of neural networks to predict similarities between two inputs, one-shot learning employs the Siamese network as neural network architecture. This architecture has been successfully used for various audio-related problems, but its use of one-shot learning in speaker recognition has received less attention. The goal of this thesis is to apply the concept of one-shot learning to classify speakers by extracting specific features, where it uses triplet loss to train the model to learn through Siamese network and calculates the rate of similarity while testing via support set and a query set to recognize the speaker accurately and faster. The proposed system is trained on the LibriSpeech dataset, which contains different audio recordings of speakers. The final one-shot is performed on few previously unseen classes, utilizing only a single sample of each type while making the classification by extracting features from training data and calculating the similarity ratio to recognize the speaker through the proposed model trained by the Siamese network. As we tested for several classes, the accuracy varied: for two classes, we got 100%, for three classes 95%, for four classes 84%, and for five classes 74%, which is significantly better than the other algorithms we tested for our solution. The results suggest that Siamese networks are a viable solution to the challenging one-shot audio classification issue.

Keywords: Audio classification, Siamese neural network, speaker recognition, One-shot learning, triplet loss.

Dedication

This report is dedicated to our adoring parents. We owe a debt of appreciation to them. Without their participation, attention, and support, we would not have gotten this far. Many thanks to them.

Acknowledgement

To begin with, all gratitude to the Great Allah for whom our proposal have been finished with no significant interference.
Also, to our Supervisor Dr. Amitabha Chakrabarty sir for his benevolent help and exhortation in our work. He helped us at whatever point we required assistance.
Lastly to our parents without their all through help it may not be conceivable. With their thoughtful help and petition we are currently very nearly our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

α Alpha

exp Exponent

$\sum$ Summation

DCT Discrete Cosine Transformation

DFT Discrete Fourier Transformation

DNN Deep Neural Network

FFT Fast Fourier Transformation

FN False Negetive

FP False Positive

GMM Gaussian Mixture Models

KNN K-Nearest Neighbors

$MFCC$ Mel-frequency Cepstral Coefficients

SNN Siamese Neural Network

TN True Negetive

TP True Positive

Chapter 1

Introduction

Human-computer interaction(HCI) approaches have been evolving for decades, and the journey continues today as new technologies emerge on a daily basis. The field of study in this area has continued to grow at a breakneck pace [88]. HCI has expanded into the areas of problem solving and decision making thanks to developments in computer technology and software design. One of the primary goals of HCI is to make computer systems more user-friendly and useful. For the interface between computer and human to work properly, speaker recognition has become a crucial aspect of this field. Speaker recognition can be described as the technique of recognizing a person based on the features of their voice. Given the speaker's distinctive pronunciation organs and speaking method, it is well established that a speaker's voice conveys personal features of the speaker. As a result, a computer may automatically identify a speaker based on his or her voice. This allows for the use of voice as a biometric in addition to fingerprint and retinal scans. In this paper, we propose an efficient and faster speaker recognition model compared to the state of the art models.

Related to the latest development in the area of smart devices, numerous research has been undertaken on how to detect the speaker in order to provide a personalized experience to its consumers. It is of great interest to the user that the machine now can interact with others in the same way as a human would. As a result, there is a rapid increase in the interest of accurately identifying the speaker with whom the machine is conversing just based on their speech in a real-world situation [36][30]. However, in such cases sometimes a new user can arrive from whom the system has never heard of. For example, a new user might call the automated customer service of a mobile operator company and ask for his account details. A service system is intended to accommodate these new users, which involves recognizing them by their voice. Traditional speaker identification techniques, on the other hand, necessitate retraining in every case when a new speaker is introduced [76]. This creates the need of storing a huge amount of data and in most of the traditional systems many voice samples at different periods from each speaker are required to accurately detect a speaker [25]. This could be problematic for a speaker recognition system. Deep learning, the branch of artificial intelligence that has had the most success in speaker recognition, faces a number of obstacles, one of which is data. To do simple tasks like recognizing audio samples of a speaker, deep learning algorithms are notorious for requiring a huge amount of training data. Also, the accuracy appears to vary depending on the words and syllables spoken. Furthermore, employing GMM

[61], Vector Quantization [42], and Deep Neural Networks (DNN) [81], substantially lower accuracies have been attained and when compelled to generate predictions on samples for which there is little supervised information, these algorithms frequently fail [47]. To overcome this barrier, we suggest a one-shot learning approach.
One-shot learning is a classification problem in which each class is given one (or a limited number of) examples, which are used to construct a model, which must then make predictions about many unknown examples in the future. Traditionally, embeddings were learned using a Siamese network for one-shot learning problems. Siamese networks that were trained with comparative loss functions performed better, eventually leading to the triplet loss function.
The proposed model's design is based on a Siamese Neural Network (SNN), a network specialized to perform verification tasks. The Siamese network is a network that has gained popularity as a result of its utilization in one-shot learning.In the early 1990s, Bromley and LeCun employed Siamese nets to handle signature validation as an image matching problem [9]. A loss function connects two identical neural networks with shared weights in the siamese network design [77]. The embedding is learned by a siamese network by reducing the contrastive loss [21] [57]. The distance between the outputs of two identical inner neural networks is used to define contrast loss. We first split our data set for testing and training. Then we conduct our training phase using SNN.
During the testing phase, our model will predict a class/person using the class's voice sample and a support set that includes some classes with only one voice sample each. Otherwise, if the system had compared the input with every sample of each class, the procedure would have taken longer. Using this approach, we focus on making our model faster and more resilient. Finding the similarities between two voice samples provides the result of recognizing which known speaker the speech input came from. Verification decisions are made by comparing the distance between the feature vectors of both samples.
The authentication model developed in this study is capable of extracting appropriate audio features as well as a mechanism that measures the similarity between the two inputs in order to identify whether the two recordings belong to the same speaker. It can be computationally costly to learn good features for machine learning applications, and it can be difficult in situations when there is limited data [[47]. This is where our system becomes really useful. Because it is capable of finding similarities between two features even with very few amounts of data. Our system is expected to function spontaneously and at a faster rate. It will be able to recognize the speaker more accurately and without causing the user any irritation.

1.1 Motivation

Voice recognition or speaker identification has been the subject of a lot of research. Near-perfect accuracy has been discovered. This sector, however, does not end there. To achieve that precision and accuracy, they must deal with large datasets and a lot of memory. As a result, the amount of time required increases. So , we wondered if we might decrease the requirement for a larger dataset, more memory, and more time while still achieving accuracy comparable to state-of-the-art voice recognition models like DNN. Furthermore, all traditional techniques test the classes that have been previously trained, but our goal was to teach the computer how to

learn who it is rather than just learning who it is. As a result, the model is able to detect classes that have never been trained or observed before. So, with the help of our thesis supervisor, we devised a novel strategy for speaker identification called one-shot learning-meta learning. Using accuracy matrix, positive similarity, negative similarity, memory, and time complexity, we will compare our model to other relevant machine learning models for speaker identification. The notion that a new field of voice recognition will be launched together with an improved resource management system prompted us to take action.

1.2 Research problem

Machine learning is the systematic investigation of scientific techniques that enable a machine to replicate human learning processes without being explicitly programmed. It also investigates biometric topographies to imitate a person's identification learning processes. Machine learning has enabled the operation of biometrics identification and has achieved significant advances in biometric pattern recognition. Various methodologies aid in the title, classification, clustering, dimensionality reduction, and recognition tasks required to develop biometric systems. The science of utilizing a person's speech as a uniquely identifiable trait is known as vocal biometrics. Speaker recognition has been one of the most intriguing and challenging in machine learning and artificial intelligence. It is utilized in the domains of human voice authentication for security and recognizing a person among a group of speakers. We can use different ways to check or recognize the particular for authentication or verification of the speaker utilizing Deep learning. According to [69], despite recent breakthroughs in speaker detection, creating single, compact representations for speech segments that we can use efficiently in noisy and uncontrolled environments remains a substantial difficulty. In [32], they introduce two novel speaker verification algorithms that employ factor analysis as a feature extractor. The first approach is built on Support Vector Machines, whereas the second relies only on the cosine distance value as a final judgment score. In the process based on SVM for verifying speakers, the score was low. Learning visual models of object categories is notable for requiring thousands of training instances, owing to the diversity and complexity of item appearance, which necessitates models with hundreds of parameters [18]. The same is true for audio classification and speaker identification, which are computationally costly and unfeasible with minimal training sets. In [27], they used audio data to train convolutional deep belief networks, which they then tested on several audio categorization tasks. Their learned features frequently equaled or surpassed MFCC features, which are hand-tailored to audio data, using a massive amount of unlabeled data. Furthermore, even when our features did not exceed MFCC, combining both allowed us to attain greater classification accuracy. However, because of a tiny dataset and expensive computation limitations, a one-shot learning strategy utilizing a Siamese network is used to tackle the problem. In [47], they investigated an approach for learning Siamese neural networks that use a unique structure to prioritize similarity between inputs. They generated good results using a convolutional architecture that outperformed previous deep learning models with near-similar performance on one-shot classification tests [47]. Having a solid understanding of similarity learning, Siamese Network is used to address neural networks [71]. Because we may train Siamese networks with minimal input, sophisticated

applications such as one-shot learning are conceivable. Using a one-shot learning strategy allows dealing with few training instances.

1.3 Research Objective

The purpose of this research is to develop a one shot learning based voice recognition system using siamese network which will be able to classify a speaker after hearing his speech. We look forward to training our dataset using Siamese Neural Network (SNN) and testing our dataset using one shot approach, which is a comparatively new approach for speaker recognition/classification problems. Speaker recognition using only a few utterances from the speaker is now widely used in a variety of fields, including

1. User verification: User verification ensures that the individual authenticating to a service is, in fact, who they claim to be for the purposes of that service. The capacity to accomplish this swiftly and without interruptions is critical to business success. For safe authentication, speaker verification has acquired widespread acceptance in the government and financial industries [18]. User verification ventures beyond traditional and physical methods of authentication to help establish identities in the digital era. Speaker recognition can be widely used to perform user verification. ID R&D's core voice biometrics product, IDVoice™ has been widely used now for speaker recognition. Our model can significantly increase its accuracy on verifying users on whom the system has not been trained yet. Our model will be able to recognize the speaker's audio sample even without having much data about the speaker.

2. Biometric authentication:Biometric authentication is a type of security measure that uses an individual's distinct biological characteristics to validate that they are who they claim to be. Biometric authentication processes evaluate physical or behavioral attributes to samples in a database that has been validated and certified as legitimate. Authentication is ensured if both samples of biometric data match. A speaker's auditory sample is a valid feature for advocating his biometric sample. Because of differences in vocal tract forms, glottis shapes, and different other organs that produce tone, no two people sound the same. Aside from physical distinctions, each person has his or her own speaking style, pronunciation pattern, word selection, and so on. Other biometrics, such as face recognition, finger prints, and retina scans, offer less potential than speaker recognition. Speaker recognition has a number of advantages over conventional biometrics, including low cost, high acceptance, and non-invasive speech acquisition. Expensive equipment and direct participation of speakers are not necessary to construct a speaker identification system. Speaker recognition has the ability to eliminate the need for a debit card, credit card, remembering a bank account password or any other security measures, and many other online services [40][75].

3. Telephone banking: Speech recognition is an artificial intelligence (AI) technology that allows software systems to identify spoken words and convert it to text. Banks are exploring security features based on AI-based speaker recog-

nition, which may automatically validate a customer's identity when they call customer care.

4. Forensic applications: Speaker recognition is commonly used to discriminate against people based on their voices. The technique of identifying if a suspected speaker is the originator of a trace is referred to as forensic speaker recognition.

5. Security: Speaker recognition, in combination with other verification methods like face recognition, could be utilized in credit card transactions as an authentication mechanism. Speaker recognition technology can be used for transaction authentication, computer access control, monitoring, and long-distance speech authentication, among other things [38]. Automatic speaker recognition, often known as voice biometrics, is a reliable approach to validate a person's unique identification for secure device access control. Instead of determining what is being said, automatic speaker recognition technology is utilized to determine who is speaking. Speaker recognition is also being utilized in conversational interfaces, messaging apps, and IoT devices ranging from smart speakers to linked cars to provide frictionless security and personalization.

6. Surveillance: Information is collected in a variety of ways by security services. Mass surveillance on telephone and radio conversations is one of them. Because this generates a large amount of data, filtering procedures must be used to discover the relevant information. One of these filtering could be the identification of relevant target speakers who are relevant to the services [38].

The i-vector representation of speech segments is employed in the most state-of-the-art solutions [32], which has resulted in significant improvements over earlier Gaussian Mixture Model-Universal Background Models (GMMUBMs) [14]. However, the majority of research relies on the user speaking a keyword to activate the equipment that will recognize them (text-dependent speaker recognition) [66]. Communicating with the devices becomes tedious as a result of this. We are looking forward to presenting a one-shot learning approach that will allow us to train the model faster and with fewer samples. So that, we don't need to store as much data for our system to detect the speaker, while simultaneously enhancing the system's accuracy in determining the likeness between the speaker audio classes.
In this paper, we emphasise on teaching our system to perform with better accuracy in the scenarios when the system has no prior knowledge of the context of the speaker's utterances. Our tests are conducted under difficult but realistic conditions with limited training data.

1.4 Thesis structure

Chapter 1 - Motivation, research problem, and research objectives are discussed in this chapter.
Chapter 2 - Literature review, algorithms were discussed in this chapter.
Chapter 3 - Data preprocessing, model interpretation and implementation were discussed in this chapter.

Chapter 4- Module, dataset, feature, algorithm analysis were discussed in this chapter.
Chapter 5 - Experimentation was discussed in this chapter.
Chapter 6 - Result analysis for the proposed model and other models were discussed in this chapter.
Chapter 7 - Future work and Conclusion were discussed in this chapter.

Chapter 2

Background

Artificial intelligence, in conjunction with cognitive science, is quickly expanding. It entails the creation and execution of an assortment of continuous applications, like discourse acknowledgment, decision-making, face identification, and DNA analysis. Voice biometrics have recently been used to verify individual identity.
Because of its simplicity, distinctiveness, and universality, the human voice is the most useful mode of communication. The following are the advantages of speaker identification over other biometric verification systems:

1. Voice is easily available, simple to use, and inexpensive.

2. Voice is very easy to obtain, and consumers can recognize persons much more easily.

We are aware of the various types of Deep learning classifiers, which require a large amount of data to train a model for the comparison between two unseen datasets and determine the similarity ratio. In many cases, when there is a limited amount of data available, this classifier becomes computationally expensive and challenging. There are, however, approaches that can work efficiently and accurately with a small number of training samples.

2.1 Literature Review

Speech recognition systems must be robust to extrinsic fluctuations caused by a variety of acoustic parameters such as transmission channel, speaker variances, and background noise since they must operate under a wide range of settings. Most speech applications use a digital filter to improve classification performance, in which the clean utterance estimation is learned by sending noisy utterances through a linear filter. With this methodology, the topic of commotion decrease turns out to be the manner by which to make the ideal channel that can adequately dispose of clamor while saving important data. As a result, various academics are looking at how to reduce the effects of environmental noise so that voice signals can be correctly classified. Lim, et al.[4], for example, introduced spectral subtraction, which adds a little amount of background noise to the voice signals. Those components in voice signals that are equivalent to noise will be hidden. However, spectral subtraction might cause the loss of some valuable characteristics by destroying various spectral components in the original voice signal [19].Support vector machine (SVM) [13]

divides speech features into several classes in order to reduce the disparity between speech features of the same class and improve classification accuracy. Nonetheless, this strategy frequently necessitates a large number of training utterances and is ineffective for applications that require a quick response.
Speaker identification research has a lengthy history. Researchers began studying speaker identification in the 1940s by using the spectrogram, but the results were unsatisfactory. Speaker recognition entered the period of artificial intelligence study after numerous computer scientists proposed the concept of artificial intelligence in 1956 [2]. Researchers planned to use computer programs to identify speakers at the time, but due to limited computer hardware capabilities and the immaturity of related algorithms, speaker identification research did not provide promising results. Machine learning [46] study began in the 1980s as a powerful branch in the science of artificial intelligence, using algorithms to evaluate data, extract useful feature information from it, and then make decisions and predictions to address the problem. Machine learning, unlike traditional software programs that are built to handle specific tasks, is learned with a huge chunk of data and then uses various algorithms to understand how to complete tasks from the data. The study of speaker identification has progressed significantly.During this time, the speaker ID technique regularly incorporates the means of discourse information preprocessing, highlight extraction, speaker demonstrating, and scoring. Pre-emphasis, framing, and windowing are all examples of speech preprocessing. Component extraction is the method involved with separating from voice flags the trademark factors that viably describe the speaker's characteristics. Mel-recurrence cepstral coefficients [29], straight forecast coefficients [23], direct expectation cepstral coefficients [34],and different elements are normal. From that point forward, speaker models are made by breaking down the recovered components. Many methodologies, for example, gaussian blend models (GMM) [10], GMM-general foundation model [70], stowed Markov model [24], and neural organization model, are as yet used for speaker displaying today.
Neural networks have been employed in speaker recognition tasks over the past ten years, thanks to the development of deep learning [85], [86]. When given labeled data, a speaker recognition system based on deep learning extracts each speaker's deep properties and performs supervised learning in a neural network model. On the TIMIT dataset, Y. Lukic constructed a convolutional neural network [56] that achieved an accuracy of 97.0 percent, corresponding to 19 misidentified speakers out of 630 [56]. Simultaneously, LSTM networks [12] and classic RNN [39] have been used to accurately predict and label sequences in a variety of tasks.A classic RNN model has a significantly lower perplexity than standard n-gram models in language modeling [28]. The LSTM network solves the long-term dependencies problem in regular RNNs and alleviates the vanishing gradient problem by inventing various gate topologies. In language detection tasks, the LSTM network currently outperforms classical RNN [53]. With regards to learning setting free and setting delicate dialects, LSTM models beat traditional RNN models [16].LSTM networks combined with a connectionist temporal classification (CTC) layer and trained using unsegmented sequence data have been found to outperform a hidden Markov model (HMM)-based system for online and offline handwriting recognition [22]. For grapheme-based speech recognition, similar strategies using a deep LSTM network have been presented [26]. In a multi-stream framework for continuous conversational voice recognition, LSTM networks have also been proposed for phoneme prediction

[35].However, the LSTM network training has four times the number of parameters as a regular RNN, making it simple to overfit. GRU has become a popular RNN variation by further optimizing the structure. In terms of performance, the GRU network is similar to the LSTM network, but the former is faster to train and less prone to diverge. LSTM and GRU are frequently used in current deep learning systems for applications including visual localization [83], traffic noise prediction [87], stock market projection [68], and air pollution forecasting [84].

As previously stated, most current research approaches are based on CNN or RNN because the speech signal is a continuous one-dimensional time series. These networks do well on various datasets [62], but no attempt has been made to combine them to research the speaker identification job. The spatial features of voiceprint and CNN are useful for spatial feature extraction and modeling spectral correlations in acoustic features due to the spectrogram that a speech signal provides [67]. Deep networks, on the other hand, can better capture extended utterances than shallow networks [64], hence stacked GRU layers for frame-level feature extraction are utilized in the model, as they have been shown to be successful for speech processing [52][51]. The proposed model, which combines GRU and CNN, should produce a better outcome.
Listeners' ability to identify and verify speakers has been examined for a long time. An early reference [3] is from 1966, for example. The voices of ten male speakers were recorded and played back to 16 audience members who knew about the speakers.The accuracy was 98% for short sentences (normal span of 2.4 seconds).When it comes to speaker recognition exams,precision alludes to whether the audience effectively recognized the speaker. It makes no distinction between how many words were correctly detected and how many were incorrectly recognized. For disyllables, speaker recognition accuracy was 87 percent. It was 81 percent for monosyllables, 63 percent for consonant-vowel excerpts, and 56 percent for vowel excerpts.As a result, they demonstrated that as the quantity of phonemes reduced, so did the accuracy of the identification.
The authors employed 45 voices in [7], including Johnny Carson, Bob Hope, Lawrence Welk, and others.Subjects were tried on their capacity to perceive the voice when it was played ordinarily (forward) and when it was played in reverse (time-switched).The forward rate had a recognition rate of around 71%.When it came to backward speech recognition, the common consensus was that it was more difficult than normal audio (about 12 percent decrease in performance). A few voices, then again, could be perceived nearly too when played in reverse, however others experienced a huge decrease in execution when played in reverse. The researchers of [7]utilized the indistinguishable voices of celebrities to examine the impacts of time-modified sound by growing or compacting it.The outcomes were comparable in that specific voices could be handily perceived even in the wake of being time-adjusted, while others could not.However, they arrived at the resolution that they couldn't anticipate what voices could in any case be recognized after the rate change or which attributes were significant for acknowledgment.
[6]utilized a sensibly enormous gathering of 24 moderators and 24 audience members in their review. The objective was to investigate how sound coding (explicitly, direct prescient coding) influences speaker commonality execution. Although the exhibition of the coded sound was less fortunate than that of the first sound, a ton

of speaker data was as yet saved. The acknowledgment rate for the crude information was 88%, and for coded discourse it was 68%. Another study [5] looked at speaker recognition with five female and five male speakers to identify. The mistake rate for great quality discourse was 23%, whereas it was 48 percent for telephone voice and 46 percent for LPC speech. The voices were chosen from a bigger selection for this study and may not have been familiar to the listeners. To allow for comparison with the stimuli, a reference sample was provided.

The issue with natural speaker recognizable proof examinations is that assembling a major gathering of individuals who know about each other is testing. As a result, most trials are limited to roughly 10-15 speakers. Speaker verification, which is another human listening task, is a distinct technique. The subject is given two voices for this exercise and must determine whether they are the same or distinct speakers. Typically, the voices offered to the listener are unfamiliar. A significantly bigger set of human responses can be obtained for this type of investigation.

The study [15] contrasted human listeners' speaker verification to that of machines. Listeners were assessed on 21 test samples with a three-second duration after being trained on a speaker. There were 65 people that listened in and 144 people who spoke to the target audience. The listeners made a total of 48,972 decisions, choosing whether the speakers from the preparation information and the 3-second test were something similar or unique.Human audience members were tantamount to machine algorithms in terms of resiliency to changing settings, although humans were more resilient. Customers, imposters, and mimics were used to verify speakers in a previous reference.

One study looked into where in the brain speaker identification and verification take place. The article [8] shown that the acknowledgment of recognizable voices (speaker distinguishing proof) and the segregation of new voices (speaker confirmation) are two separate cycles that occur in various spaces of the mind.Patients with one or the other left or right cerebrum injury were remembered for the review. A deformity in the segregation scores was brought about by a sore in the left side of the equator of the mind (speaker verification).

There is a huge group of writing on speaker identification, with two major approaches: classification and verification. Techniques of classification train one model with an output limited by the classes (or, in this case, known speakers) with which it was trained.When a new speaker enters the scenario, the model attempts to match it with one of the previously known speakers, which results in a false positive. Verification techniques, on the other hand, train a model for each known speaker, with their outputs indicating the likelihood that the speaker is the one for whom the model was trained.This type of technique can register if a speaker is unknown (when all models provide low probabilities), but it is generally less accurate than classification techniques [11].

[31] identifies speakers by contrasting a proportion of comparability between the sound of a speaker to be recognized and the examples recently produced for known speakers. Later, in [45] the author proposes changes to this system in which the database and architecture change, but the similarity measure remains constant.With raw audio, neural networks have also been used [74], where a CNN extracts the relevant information and an impromptu verifier is created for every speaker. The authors of [62] describe the VoxCeleb database and trained deep learning models for speaker identification and verification. The cosine distance between two signals

is used.In [73] speaker verification is performed utilizing a Siamese representation of two Long Short-Term Memory (LSTM) organizations and a contrastive misfortune. It is significant that the utilization of neural networks in conjunction with audio has not only addressed the issue of identifying and verifying the speaker, but has also addressed the issue of extracting audio features to make them more robust. [63], [58], [44], [60], and [55] use embeddings or complete neural networks to generate features that are then used by statistical methods to verify a speaker.
Depending on the database used to train, the aforementioned works achieved verification accuracy rates of more than 80%.
In the field of PC vision, single shot learning with siamese neural organizations has been considered, for instance, in transcribed person recognition[47] and object classification classification [20]. In any case, this idea has gotten unreasonably minimal public consideration in the field of human voice arrangement, where the expectation is to perceive the marks of human voice. This is in all likelihood because of the way that profound learning calculations in the space of sound are moderately new, having just been openly accessible for the past ten years. [27][50][41].
Li Fei Fei et al[17] put forward One-shot learning which utilized a Bayesian strategy to achieve picture arrangement utilizing a single shot learning.The creators wanted to construct a Bayesian model that could sum up to the new article classes with a couple examples,if any whatsoever they called their unique method Bayesian a single shot. Following this,Li Fei Fei et al.published a paper that expanded on this topic[20].Lake et al.[33] examined one-shot learning within the realm of handwritten characters,where the author looked at one-shot learning through the lens of science of the mind.According to their theory, their generative model could deduce hidden strokes in unseen characters by using knowledge from previously seen characters. From that point forward, they have distributed various papers on idea learning and a single shot order, and the omniglot dataset has been delivered as a dataset for benchmarking for picture a single shot classification [33][37][48].
The use of deep siamese cnn for image classification was investigated by Koch et al [47].Its application was to utilize one-shot learning.The authors used the Omniglot handwritten character dataset to train a siamese network model to rate resemblances between various input pairings,which was then used to infer to previously undiscovered classes.In learning generic picture features capable of extrapolating to one-shot classification, this technique based on metrics outperformed all possible baselines available at the time.They also advised that this methodology be applied to other domains' one-shot learning assignments [47]. This thought was additionally reached out by Google DeepMind's Vinyals et al. [59], who utilized a more confounded organization engineering and a preparation approach that impersonated the trial case.When compared to other competing algorithms,the authors were able to significantly improve one-shot classification on the Omniglot dataset [59]. With regards to entity relation extraction, one-shot learning with convolutional siamese networks has been investigated [65].This extraction of fine-grained relationships proposed was viewed as a one-shot classification problem,with the objective of predicting unusual relationships joining samples using a single or a few examples.The authors were able to obtain auspicious findings and demonstrate the inherent advantages of domain-specific data extraction [65].A quadruplet siamese deep neural network for one-shot learning for tracking of visual object was recently employed by Dong et al.[79]. They put forward a network design that used an aggregate of four info tests for a single

shot learning which differed from previous work.The creators joined two misfortune functions,triplet and pair loss,with the trio misfortune determined regarding three examples and the pair misfortune determined according to a solitary pair of cases. An additional weight layer was used to select the weights in combination between these losses,to simplify the process of bias correction.

A single shot learning with siamese organizations has not gotten as much perception in sound as it has in PC vision.An architecture named siamese convolutional neural network(SCNN) was designed for a single shot speaker identification which was proposed by Velez et al. [76]. in a paper distributed in 2018. The author's main objective was to develop a model to identify whether two audio recordings had been created by the same speaker, even though the speakers were completely different.The proposed system comprised of 3 elements: a nonexclusive verifier, an outside data set for known client sections, and a speaker determination framework, with the verifier deciding if two sound signs arose out of a similar speaker.The The time-frequency spectra of both the input audio signals were used to represent them in this system of general verification which was executed as a convolutional siamese network.The authors also looked at a variety of other Siamese organization geographies and provided details regarding the main three models.Using on the notable VGG [43]and ResNet 50 [54] designs, these three portrayals accomplished with a serious level of accuracy of 81.2% in certifiable ecological assessment errands, demonstrating that a single shot learning with siamese organizations can be stretched out to true sound classification issues.Elof et al. [80] investigated the topic of multimodal one-shot learning by using pairs of spoken and visual numbers to determine whether a siamese representation exists could do one shot classification on this input data in multiple formats.The authors were able to acquire improved accuracy outcomes for picture pixels and dynamic time warping over speech when compared to a nearest neighbour classifier.

Notwithstanding a single shot learning, Siamese organizations were utilized for other sound difficulties. With the use of siamese networks,Manocha et al [72] suggested a new method for sound collection based on content.The authors' approach was to use a siamese system to estimate encrypted vector representations of audio data, which could then be used to efficiently locate semantically similar sounds. The vector creation process is similar to audio fingerprinting, in which semantic information is recorded in a numerical format that is easily comparable. The findings of the authors revealed that their method could extract mutual information across audio data within the same class and could be used for a diverse array of audible retrieval tasks. Zhang et al[78] used siamese cnns in the framework of audio quest.As their method, they used a siamese framework which is able to extract characteristics as well as predicting similarities among voice limitations as well as actual audio.For this topic,the authors presented two systems:symmetric IMINET and asymmetric TLIMINET.

IMINET used two similar networks that were taught from the ground up,but TL-IMINET employed pre trained embedding channels that were created through learning algorithms from activities such as spoken language recognition and environmental noise categorisation.The discoveries showed that the two variations of the framework outflanked an advanced framework, and that utilizing the idea of move learning for a siamese organization altogether further develops sound recuperation effectiveness [78].

2.2 Algorithms

Machine learning is used in a variety of disciplines, including security, analysis, recognition, and prediction. It is the act of providing data to a computer that enables it to perform intelligently without explicit programming by analyzing past data and anticipating new data. Machine learning is also employed in our proposed idea. Our method is supervised machine learning, which is the application of artificial intelligence algorithms to discover patterns in data sets using labeled data. We used a dense layer, a Neural network layer for embedding models, and a siamese network for determining sample similarity. After that we compared our results with the traditional approaches like MLP, KNN and Random Forest.

2.2.1 Dense Layer

Every neuron in the dense layer gets input from all neurons in the former layer, making it a profound associated neural organization layer. The dense layer is observed to be the frequently utilized layer in the models. Behind the scenes, the dense layer performs matrix vector multiplication. The qualities in the matrix or the boundaries can be prepared and refreshed by means of backpropagation. As its yield, the dense layer produces a m-dimensional vector. As a result, the dense layer is predominantly used to change the components of the vector.Dense layers furthermore utilize the vector to perform tasks like scaling, pivot, and interpretation. In figure 2.1 all neurons in the previous layer, which is the input layer, provide input to n11, n12, and n13. And all of the neurons in the preceding layer, n11,n12, and n13, provide input to n21,n22, and n23. The output layer "O" produces an m-dimensional vector.

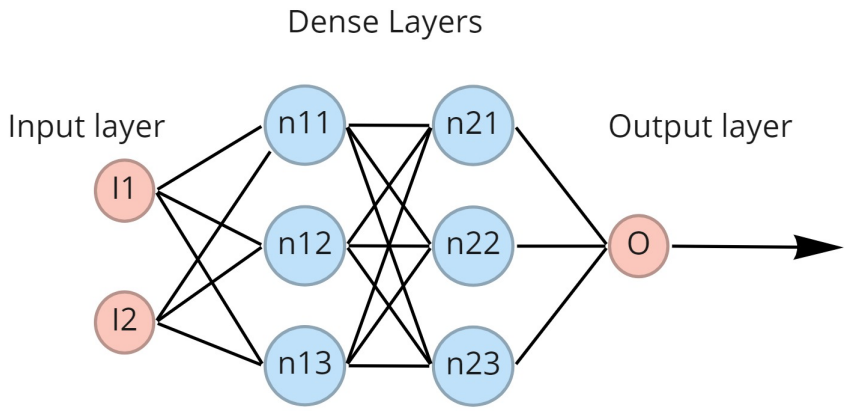


Figure 2.1: Dense Layer

2.2.2 Siamese Neural Network

A Siamese Neural Network is a Meta-Learning idea product. To understand the SNN, we must first comprehend the Meta-Learning idea. Simply defined, Meta-

Learning is a cutting-edge concept in the machine learning field that is used to train a model to learn how to learn. The Siamese Neural Network is being used to put the Meta-Learning idea into practice. Bromley et al. first presented this idea in Signature Verification, where they used a Siamese Time Delay Neural Network to compare an unknown signature to a known one [9]. This model can have two or more similar subnetworks. The weights and attributes of these subnetworks are exactly the same. The main idea behind this model is to utilize these subnetworks to compare samples of data in order to determine how similar they are. The first and second subnetworks are learning to recognize similarities and differences between two data respectively. As a result, the trained Siamese network, which is made up of two subnetworks, is able to recognize untrained input. To do this, the model is given a support set that contains several types of data. As a result, anytime the model is given untrained data, it will attempt to discover similarities between the untrained data and the support set data. The untrained data can then be classified as a class from the support set if the untrained data's class is also in the support set. Otherwise, the untrained data will be classified as unknown. In the Siamese Network figure 2.2, two subnetworks are coupled by a shared layer with identical parameters and weights.

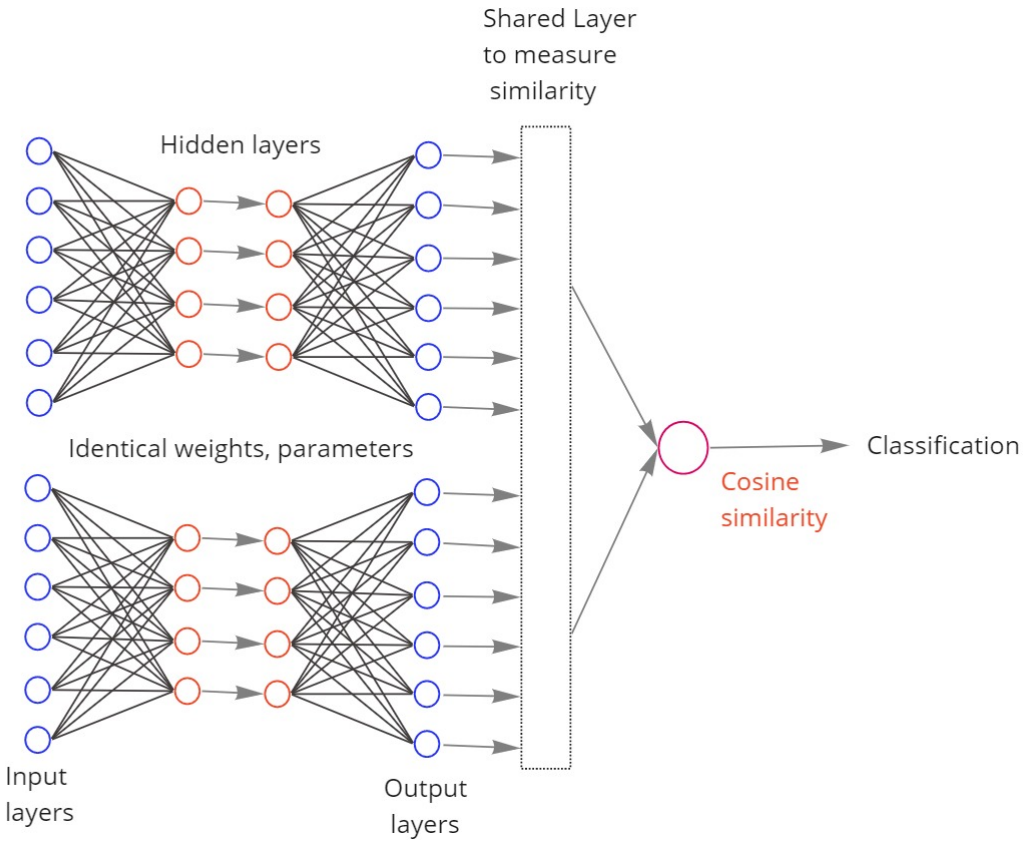


Figure 2.2: Siamese Network

2.2.3 Multi-layer Perceptron Classifier

The acronym MLPClassifier refers to a Multi-layer Perceptron Classifier, which is related to a Neural Network. It's a sort of artificial neural network that feeds back

information. The term MLP is ambiguous; it can apply to any feedforward ANN or to networks composed of multiple layers of perceptrons (with threshold activation). Unlike other different classifiers such as Naive Bayes or Support Vectors, MLPClassifier uses an underlying Neural Network to do classification. The input layer, the output layer, and the concealed layer are the three layers that make up this system shown in figure 2.3. The input layer receives the signal that will be processed. Prediction and classification are activities that fall under the output layer's scope. An arbitrary number of hidden layers added between the input and output layers serve as the MLP's real computational engine. The MLP's neurons are trained using the back propagation learning technique. MLPs can solve issues that aren't linearly separable and are designed to approximate any continuous function. The most frequent MLP applications are pattern recognition, classification, approximation and prediction [82].

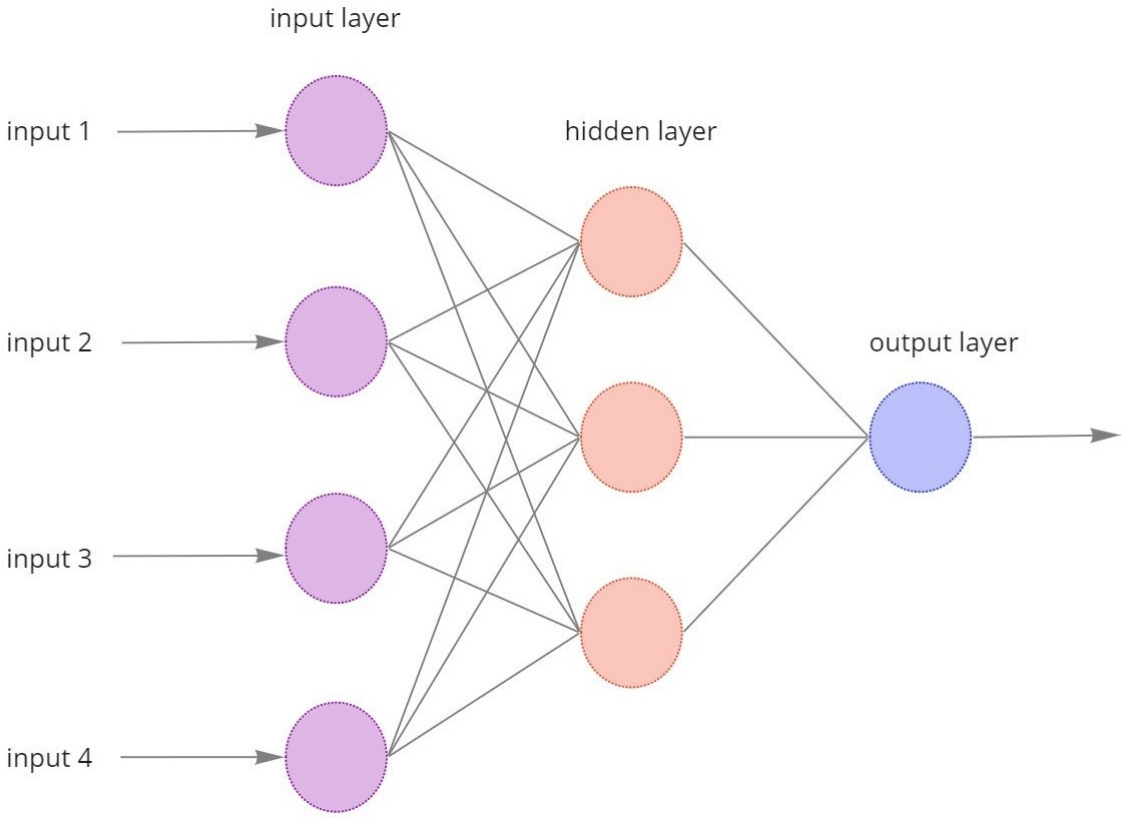


Figure 2.3: Multi Layer Perceptron

2.2.4 KNN Classifier

The K-Nearest Neighbors (KNN) algorithm is a popular machine-learning approach that has been used in huge data mining projects. It's also commonly used for graph construction and biomedical signal analysis. KNN is a straightforward, convenient supervised machine learning algorithm (it depends on labeled input data to develop a function that provides an appropriate output when given additional unlabeled data) that may be used to tackle regression and classification tasks. In k-NN classification,

the function is only approximated locally, and all computation is postponed until the function is assessed. Because this technique depends on proximity for classification, normalization of the training data can improve performance significantly if the features reflect distinct physical units or scales. Assigning weights to the inputs of the neighbors, so that the near neighbors contribute more to the average than the farthest neighbors, is a successful technique for both classification and regression. Similar items, according to the KNN algorithm, are near together. To look at it another way, objects that are connected are closer to one another.

2.2.5 Random Forest Classifier

Random forests is a supervised learning method that creates outstanding outcomes in most circumstances with or without hyper-parameter adjustment. It's a machine learning algorithm that's both flexible and simple to use. It is capable of being used for both classification and regression. It is also one of the most extensively utilized algorithms owing to its efficiency and flexibility. It combines several decision trees to generate a more precise and stable estimation. While dividing a node, the accurate assessment from a randomized subset of features is chosen instead of the most significant characteristic. As a consequence, there is a wide range of options, which contributes to a superior model.

Chapter 3

Proposed Methodology

The goal of our proposed model is to analyze a person's speech and try to identify them. Although, a major feature of our model is the ability to recognize a person's speech that it has not been trained on and must identify with the assistance of only one sample of a class from a support set. As an outcome, the model is capable of achieving one-shot learning.
The following are some of the major steps of our research:

1. Preprocessing of audio data at the initial stage. As our model trains and tests data differently, the audio data is preprocessed and the data are organized in a certain structure.To pass the audio data through the model, we need to prepare it so that our model can extract features and get trained accordingly.

2. The features for the training data are then extracted and put into the model one by one. In addition, the model is trained using the whole training data set. The model must be taught how to learn, which is one of the main aims of the training stage. In other words, the model learns to detect a class/person using audio data that is never taught during the training phase.

3. Finally, the prediction phase starts, in which the testing data, which the model has never seen/trained, is given to the model to predict using a support set. So, the outcome of the testing phase is that the model will recognize a class/person whose voice has never been trained in the training set. The workflow of our research has been shown in figure 3.1 below:

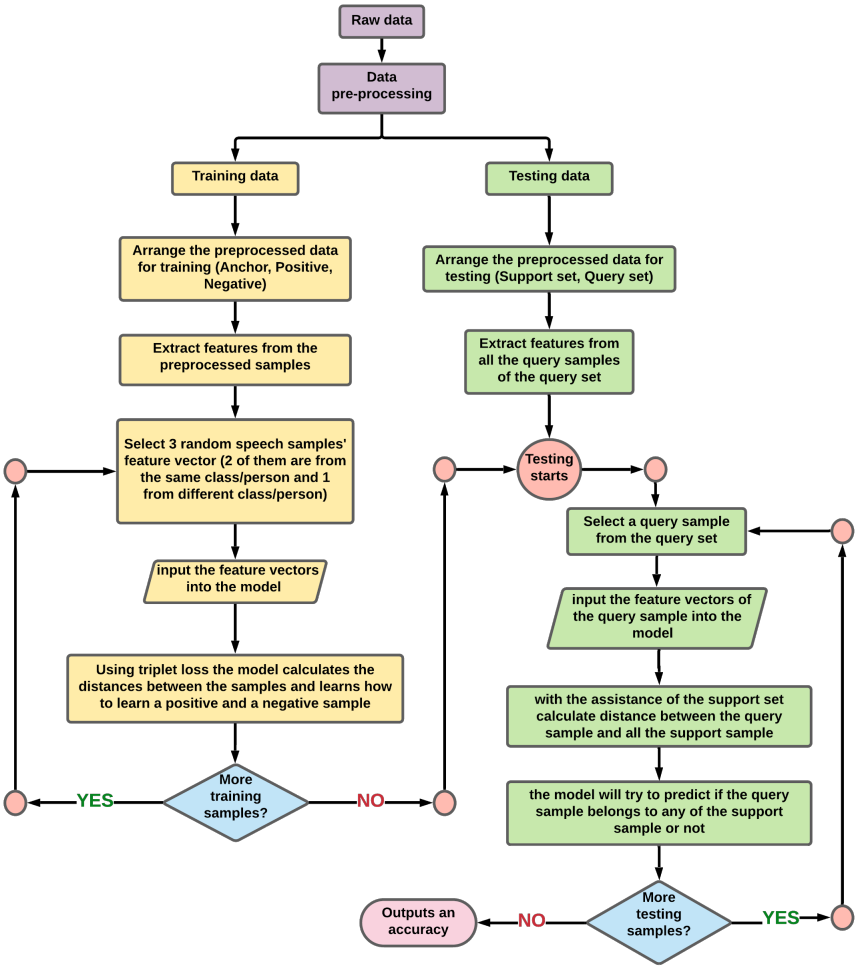


Figure 3.1: Workflow of our research

3.1 Data preprocessing and data arrangement

Audio samples are commonly displayed as a time series, with the amplitude of the waveform as the y-axis measurement. The amplitude is usually calculated as a function of the pressure change around the microphone or receiver device that picked up the audio in the first place. These time series signals will frequently be our only input data for constructing a model unless our audio samples have metadata linked with them. MFCC is one of the most useful characteristics for extracting information from audio waveforms (and digital signals in general) has been around since the 1980s. Dataset preparation, feature extraction, and feature engineering are actions we take to extract information from the underlying data, information that may be used to predict a sample's class or the value of a target variable in a machine learning setting. This approach is mostly dependent on identifying components of an audio signal that can assist us in distinguishing it from other signals in audio analysis.

3.1.1 Noise remove

For attaining One-shot learning, the data preprocessing and arranging structure is a bit challenging and unique. The audio data must be cleansed for excessive noise at the preprocessing stage. The inclusion of noisy data in a data set can significantly lower the accuracy of any valuable information prediction. Many research studies have shown that data noise significantly reduces accuracy and leads to poor prediction. As a result, removing noise from the data is critical. We have already chosen a cleaned dataset, so no noise is found when screened.

3.1.2 Feature extraction

MFCCs are still the most sophisticated method for extracting data from audio samples. Despite the fact that libraries like Librosa provide a one-liner in Python to compute MFCCs for an audio sample, the underlying math is a bit difficult. The following are the steps to calculate MFCCs for a given audio sample:

1. Shorten the signal by slicing it into frames (of time).
2. Calculate the power spectrum periodogram estimate for each frame.
3. On total the energy in each filter, apply the mel filterbank to the power spectra.
4. Take the log filterbank energies and apply the discrete cosine transform (DCT) to them.

Equations for MFCC:

$$S_i(k) = \sum_{n=1}^{N} S(n)h(n)e^{-j2\pi kn/N} \qquad 1 < k < K \tag{3.1}$$

$S_i(n)$ is the time domain signal in (3.1). We'll have $S_i(n)$ once we frame it, where n is the number of frame samples. The complex DFT is then calculated, and $S_i(k)$, where I is the frame number corresponding to the time-domain frame, is formed. The hamming window is $h(n)$ in this dicreate fourier transformation of the frame. The length of DFT is K.

$$P_i(k) = \frac{1}{N}|S_i(k)|^2 \tag{3.2}$$

The power spectralm of frame i is determined using a periodogram in (3.2). Here, we square the result of taking the absolute value of the complex fourier transform.

$$m = 1125\ln(1 + \frac{f}{700}) \tag{3.3}$$

$$m^{-1}(m) = 700 * (\exp\left(\frac{m}{1125}\right) - 1) \tag{3.4}$$

(3.3) and (3.4) are used to convert frequency to mel and mels to frequency, respectively.

$$H_m(k) = \begin{cases} 0 & k<f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k>f(m+1) \end{cases} \tag{3.5}$$

The first filter bank (equation no 3.5) will start at the very first point, peak at the next, and then go back to its initial point at the final point. The second filter bank will start at the second point, peak at the third point, then drop to zero at the fourth point, and so on. The formula for estimating these is as follows. Here's a list of M+2 Mel-spaced frequencies, along with the set of filters that is needed.

3.1.3 Glottal pulse remove

The glottal pulse is that part of a voice which is not needed for speech processing. So, we have to separate the glottal pulse from the vocal tract. The vibration of the vocal chords (the source) is convolved/filtered by formant resonators linked with the vocal tract in a speech signal (the channel). To do this, we use a log function.

$$x(t) = e(t) * h(t)$$

$$\begin{aligned} \log(x(t)) &= \log(E(t) * H(t)) \\ &= \log(E(t)) + \log(H(t)) \end{aligned}$$

$$\log(E(t)) + \log(H(t)) \rightarrow h(t)$$

$$x(t) = h(t) \tag{3.6}$$

Here, x(t) in 3.6 represents speech, e(t) represents glottal pulse, and h(t) represents the vocal tract, which we only obtain after applying a log and low pass filter.

3.1.4 Data arrangement

Furthermore, for the model to perform efficiently, the training and testing data must be organized in a certain way. The primary distinction between traditional learning and one-shot learning approaches is how data is trained and tested. In a one-shot learning approach, the training data of the model may be accomplished in two ways. The pairwise similarity and the triplet loss are two examples. With our model, we will work with the triplet loss which basically uses the enhanced version of pairwise similarity.
In order to work with triplet loss, we need three audio samples per training. We separated our training data into two directories. One directory contains anchor samples, whereas the other has positive samples. Here, the anchor audio samples are the base of the training. It is one audio sample of a class/person. And the positive

samples are the same class/person's another sample. The audio files include speaker id in the name, which we split and placed into a numpy array in order to match the anchor class and positive class. So, person A's anchor sample is at the anchor directory and file named 1.flac, the same person's positive sample is at the positive directory named 1.flac as well. This is how the anchor and positive samples are kept in track.Then, in each iteration, we took negative files from various speakers so that the speaker ids don't match. As, the negative samples are randomly stored, so with the anchor audio sample 1.flac, a different person's audio sample is picked from the negative directory thus creating negative samples. During the training phase, one anchor sample and its matching positive sample will be chosen, followed by a negative sample. Three samples are chosen each time.
Now, for the testing phase, we reorganized the testing data in a unique manner. In a one-shot learning approach, the testing is done by a query set and a support set. So, we separated the testing data, and made a support set and a query set. The query set contains audio samples which the model never trained on and the support set contains one audio sample of few classes/persons which is also not trained by the model. Using one query sample and the support set sample, our model tries to predict if the query audio sample's voice belongs to any of the class/person from the support set.

3.2 Model Interpretation and Implementation

The siamese network is used in our model to achieve improved predictions using only a few numbers of audio samples. A siamese network is an artificial neural network that consists of two or more identical networks, and by identical it means they share the same weights and parameters across the network.
The concept of meta-learning is the main idea that a siamese network achieves. Meta learning is a subset of metacognition that focuses on understanding one's own learning and learning mechanisms. The training of a competent machine learning model often necessitates a huge amount of data. Humans, on the other hand, learn new concepts and skills considerably more quickly and efficiently. Meta-learning aspires to achieve comparable results: acquiring new concepts and skills quickly with a limited number of training instances. We expect a good meta-learning model that is capable of adjusting or generalizing to new activities and situations not encountered during training. During the test, the adaptation process, which is basically a small learning session, occurs with very little experience of the new task configurations. The customized model will eventually be able to fulfill new tasks. Simply described, meta-learning is the notion of learning how to learn. And, by realizing the concept of meta-learning, the one-shot learning approach has become a reality, in which a neural network can identify someone using only one sample of their picture or speech. Meta-learning and adaptation are the two phases of the meta-learning process. During the meta-learning phase, the model slowly learns an initial set of parameters across tasks; during the introduction phase, it concentrates on efficient knowledge to acquire task-specific attributes. Meta-learning is also known as learning to learn because it occurs on two levels. The method that we adopted is the triplet loss method of training.
The triplet loss is defined as:

$$L(A, P, N) = max(|f(A) - f(P)|^2 - |f(A) - f(N)|^2 + margin, 0) \tag{3.7}$$

The distance between samples is calculated using the triplet loss function (equation no 3.7), which takes three inputs per training. The anchor sample, which can be any unique data point, is the initial input. The positive sample, which belongs to the same class as the anchor sample, is the second input. The third sample is a negative sample, which is a sample that does not belong to the same class as the first two. Here, a class represents a person, and a sample of a class is that individual's audio sample. For training the model, If the anchor point is an audio sample of a person, another audio sample of that person is picked for the positive sample, and an audio sample of a different person is picked for the negative sample. However, in order to accomplish one-shot learning, we can only have one sample per class in the support set while testing the model.

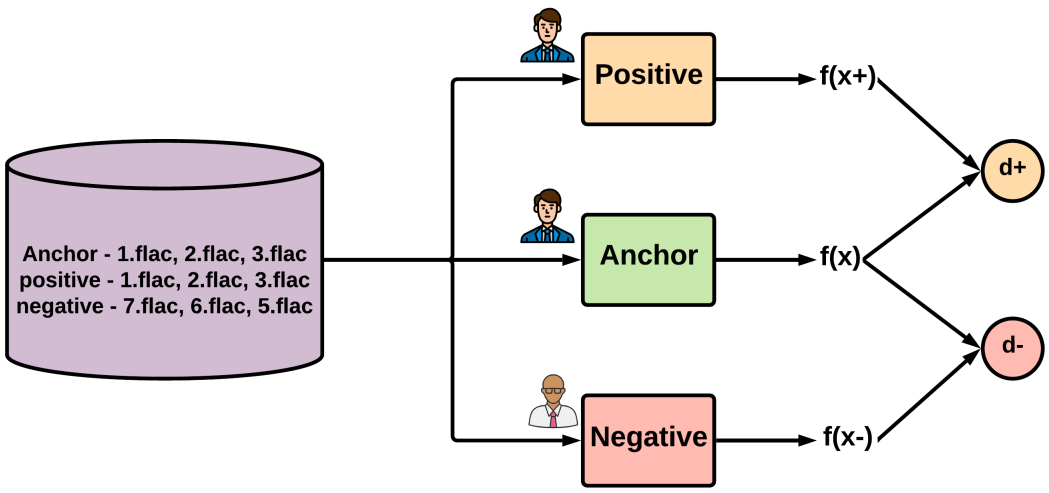


Figure 3.2: Model Training.

In figure 3.2, the 1.flac file in the anchor directory and the 1.flac file in the positive directory are both audio samples from the same class. The 7.flac file in the negative directory belongs to an audio sample out of a different class. Here, the negative directory is randomized, whereas the anchor and positive directories are numerically ordered.Basically, we choose three audio samples at a time to train the siamese network. Each time the siamese network is trained, an anchor sample and its matching positive sample, as well as a randomized negative sample, are chosen, as indicated in the figure 3.2 For example, 1.flac audio sample is picked as an anchor sample from the anchor directory. As the positive sample must be from the same class of the anchor sample class, the model will choose 1.flac file as the positive sample from the positive directory as well. Then, the model will pick a randomized sample which does not belong to the same class as the anchor class. Here, the naming and arrangement of similar classes audio samples are done in the data preprocessing step. After the selection part of the input audio sample, the audio samples are sent into

the neural networks. Although there are three neural networks illustrated in the diagram, there is only one shared neural network in the model which processes all three sample inputs with the same weights and parameters. As a result, it is known as a siamese network. Next, from the input audio samples, the neural network derives three feature vectors from three audio samples. After that, the model estimates the distance between the anchor and the positive sample ($d+$) and the distance between the anchor and the negative sample ($d-$). Following the calculation of distance, the feature vectors of the same class should be close together, while the feature vectors of different classes should be well separated as shown in the figure 3.3. The negative distance is much larger than the positive distance. In the loss function, a positive tuning hyper-parameter, alpha(α), is employed to ensure that the negative distance is substantially bigger than the positive distance. The classification is accurate and the loss is zero if d- is larger than d+ by an alpha margin. If d- is not greater than d+ by an alpha margin, then the classification task is incorrect, and some loss value will arise.

1. If $d- < sum(\alpha, d+)$, then loss= $(d+) + \alpha - (d-)$, we want the loss value to be small.
2. If $d- >= sum(\alpha, d+)$, then no loss.
3. Loss function, $L = max(0, [(d+) - (d-) + (\alpha)])$

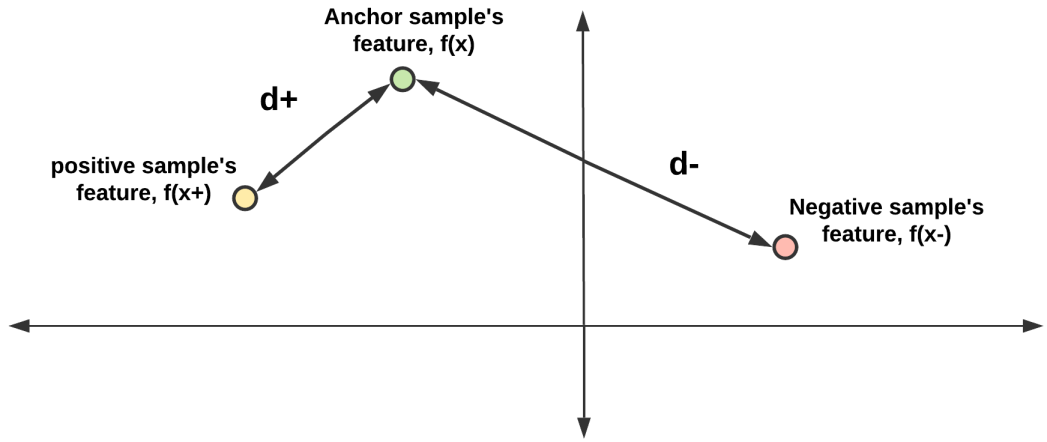


Figure 3.3: Feature vector space

The loss function is called the triplet loss function which uses three samples and with positive and negative samples, calculates a distance between the samples. Here, if $(d+) - (d-) + (\alpha)$ is greater than 0 then there is some loss and the classification is not correct. Conversely, if$(d+) - (d-) + (\alpha)$ is less than 0 then we can say that there is no loss and the classification is correct. Now, the model computes the derivative of the loss in contrast with the model attributes and then uses gradient descent to update the attributes of the model using the loss function. Until there is training data to work with, the model will keep adjusting the parameters to lower the loss function values.

The most unique feature of our model is the testing phase. Our model achieves one-shot learning due to its uniqueness in testing. The major part of testing of the testing phase is the comparison between the support and query set. The query set and the support set are arranged from the testing data in the data preprocessing stage. One key difference between one-shot learning and other traditional machine learning approaches is that the testing data is completely fresh to the model, and it has never been trained on any of the support set samples.

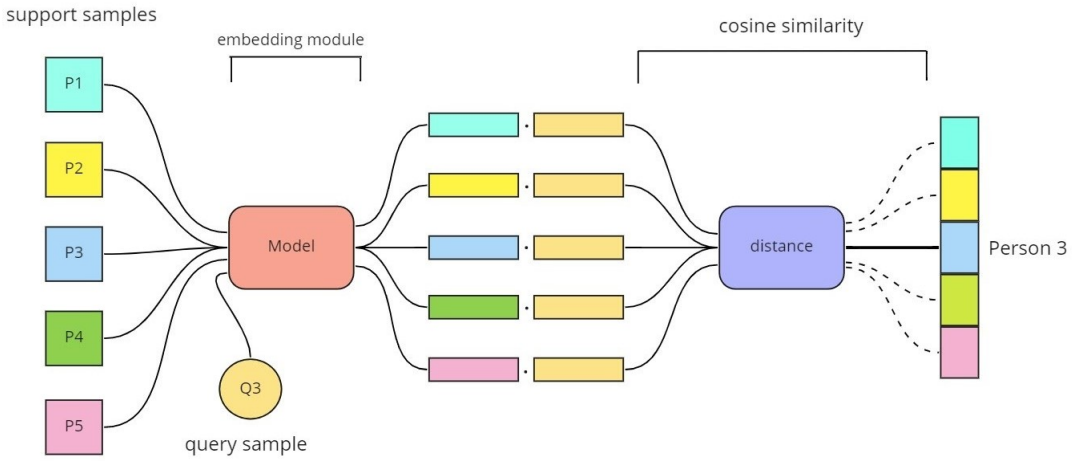


Figure 3.4: Testing phase explained

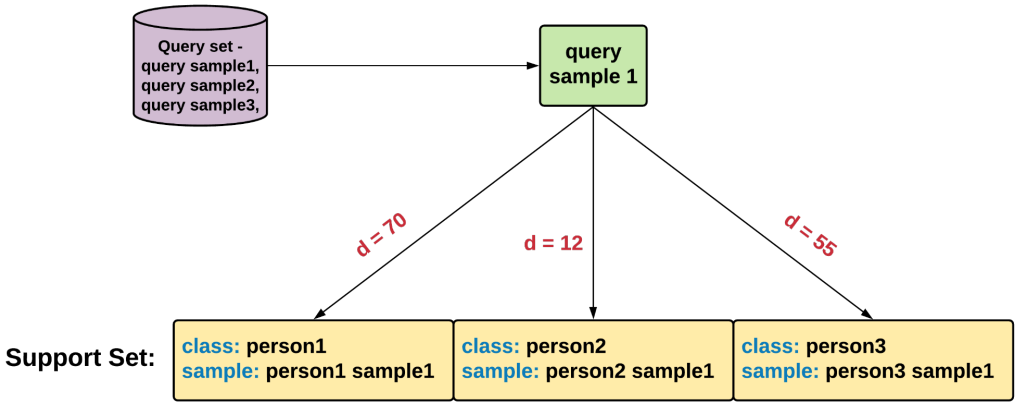


Figure 3.5: Testing phase of the model

Here, the query set contains audio samples of class/people who were never trained during the training phase. The support set also includes a small number of classes, each with only one sample. These individuals in the support set are also the ones that have never had their voices trained by the model. The reason this approach is called one-shot learning is because the model has to predict a query sample by comparing it to only one single sample of an individual or class. Intuitively, the model is hearing a person's voice for the first time(the query sample), then hearing

one voice sample each from some individuals or classes(the support set), comparing the query sample to each voice sample from the support set, and then attempting to predict which person's voice sample from the support set matches the query sample best. If, instead of obtaining one voice sample from each person, we take a few voice samples from each individual, then the learning approach is no longer referred to as one-shot learning. The approach will subsequently be considered as few-shot learning. The higher the number of samples the easier it is for the model to predict. In figure 3.4, there are 5 support samples and 1 query sample which we want to recognize. We are passing it through the embedded model and after calculating the cosine similarity we get the desired output. Also, there are three classes illustrated in the figure 3.5. Therefore it is known as three-way-one-shot learning. If there are n numbers of classes in the support set, then it is called n-way-one-shot learning. The higher the number of classes the difficult it is for the model to compare and predict. Moving on to how model testing is carried out, the model starts testing by selecting a random query sample. Then, the query sample is sent into the neural network to extract features, which are then compared to each of the support set class samples whose features are also extracted by the neural networks. Following the comparison, a distance value is computed between the query sample's feature vector and each support sample's feature vectors. Finally, the model chooses the support class with the least distance value with the query sample as the predicted output. So, the predicted person/class is the support set class whose sample has the smallest distance value in the feature vector space. As we can see in figure 3.5, query sample 1 has distances 70, 12, 55 with classes person1, person2, person3 in the feature vector space respectively. As the distance between query sample 1 and person2 sample1 is the least among all other samples, the model thereby predicts the query sample 1 to be the voice of person 2. Using the same technique, all remaining query samples are also evaluated with the support set samples. Although the model has never heard the voice of person2, it still can effectively recognize it. Therefore, the concept of meta-learning and the one-shot learning is achieved by this model.

Chapter 4

Preliminary Analysis

This section describes the preliminary study of the proposed model and the dataset for the speaker recognition task. We'll establish the suitable performance measures for our situation, explain why they're appropriate, and discuss what other options we've considered. Our detailed analysis has led us to the conclusion that we should break down our overall task into three stages: gathering and preprocessing relevant data, training our model using Triplet Loss, and finally testing our one-shot learning strategy. Keras, a neural network library that leverages Tensorflow as its backend, was used. Also, we used Librosa, a Python module, to read and preprocess our dataset.

A hypothetical comparison based on alternative models and training processes will also be shown. It will provide an overview of the research issue, challenges we've encountered while working through several methods, and how we came up with the concept to use the meta learning methodology "one-shot learning."

4.1 Module analysis

The proposed model for one-shot voice recognition requires two types of python module support.One is Keras, an open source Python framework for creating and testing deep neural networks that is both powerful and simple to use. It introduces TensorFlow, one of the most efficient numerical computing frameworks for designing and training neural network models only with a few lines of code. It was made with the intention of facilitating quick experimentation. It offers uniform and straightforward APIs, reduces the amount of user activities required for common use cases, and gives clear and actionable information in the event of a user error. It enables for quick prototyping and operates on both the CPU and GPU. Keras' basic data structures are layers and models. The most fundamental type of model is the sequential model, which is a linear stack of layers. We will use the Keras functional API for more complicated designs, which allows us to build arbitrary layer graphs or write models entirely from scratch via subclassing. Keras is also a very adaptable framework for iterating on state-of-the-art research concepts. Keras maintains the notion of progressive disclosure of complexity, making it simple to get started while also allowing it to handle arbitrarily complicated use cases with just incremental learning required at each stage.
For our proposed technique another one is, Librosa, which is utilized to read, pre-

process, and extract data from the dataset. Librosa is a Python module for signal processing in audio and music. Librosa provides high-level abstractions of a wide range of common functions used in the domain of music information retrieval [49].The librosa.core submodule contains a number of useful functions. The four primary areas of basic functionality are audio and time-series operations, spectrogram computation, time and frequency transformation, and pitch operations.
The librosa.feature module supports a wide range of spectrum representations based largely on the Fourier transformation in a short time. Audio signals are usually represented by the Mel frequency scale because it gives an abrupt model of human frequency perception [1]. It's a Python module for analyzing audio signals in general, but it's oriented at music in particular. It comes with everything we will need to put together a MIR (Music Information Retrieval) system.

4.2 Input Data Analysis

Vassil Panayutuv and Daniel Povey obtained almost 1000 hours of speech data for the LibriSpeech dataset, which was discussed in the preceding chapter. The audio data is collected from the LibriVox project's audio books. It's utilized in a range of applications, including speaker recognition and automatic speaker verification.We had difficulty acquiring a well-organized dataset for our methodology considering our methodology is new in the field of voice recognition.
The dataset is divided into twelve folders . However, it allows users to selectively download sections of it based on their needs. Subsets having the word "clean" in their name are said to be "cleaner". Based on our requirements, we downloaded four subsets, "Train-clean-100", "Train-clean-360", and "Test-clean". The remaining subgroups will be evaluated as a part of future work. Once downloaded, all the folders contain information about the speakers, books, and chapters written in text files, as well as multiple samples of more than a thousand speakers with various speeches. However, we only require the speeches. Although there are multiple folders arranged, we still needed to modify our dataset for our proposed model, which we have discussed in the previous chapter of our report.

There are 921 speakers in Train-clean-360, 255 speakers in Train-clean-100 and 40 speakers in Test-clean, shown in table 4.1 and figure 4.1

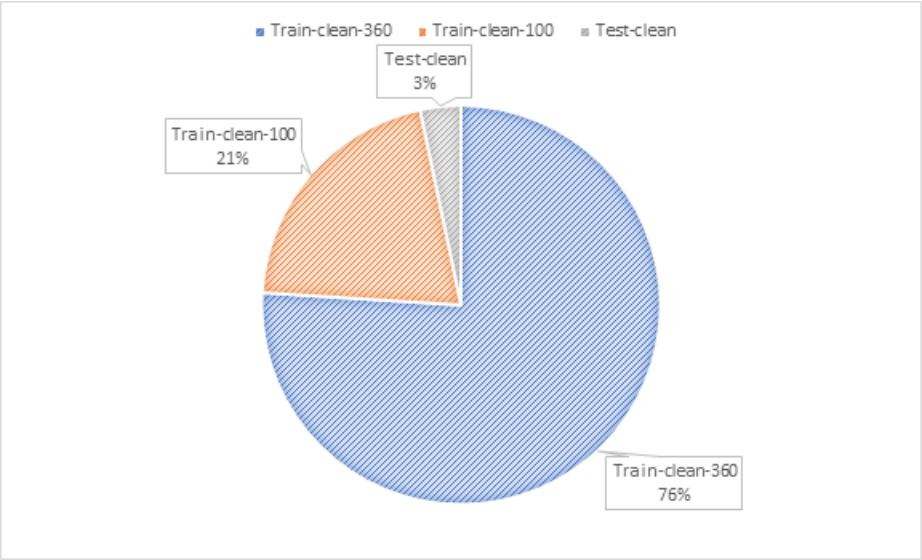


Figure 4.1: A pie chart of Data summary

As a result, there are 1176 classes or speakers in total in the training class. For each anchor and positive array, we took a total of 33675 samples. Then we took 33675 samples for the negative array to match the number. In our research, a total of 101,025 samples were trained as shown in figure 4.2.

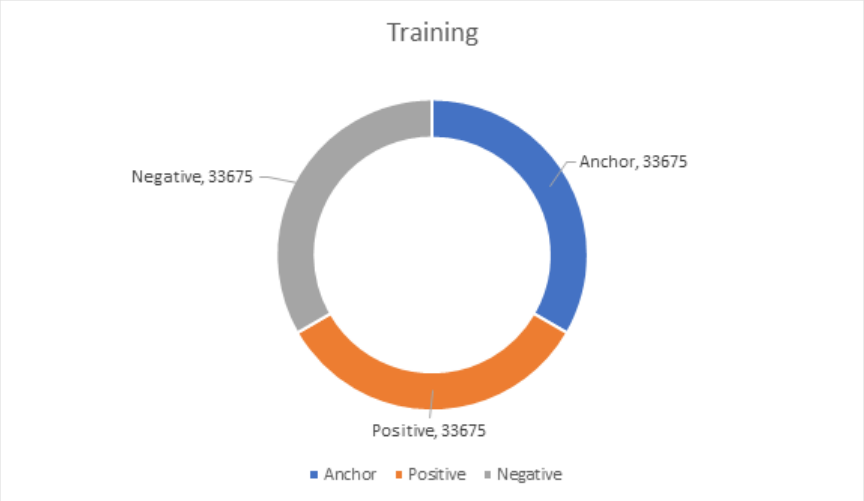


Figure 4.2: Training Data summary

Subsets	Speaker	Phase	Arrays	Samples
Train-clean-360	921	Training	Anchor Positive	33675 33675
Train-clean-100	255		Negative	33675
Test-clean	40	Testing	Support Set Query Set	1 for each class n for each class

Table 4.1: Number of samples in each sets

According to our research, Librispeech has performed well with traditional approaches. As it is supposedly cleaner than other most used datasets and also works better on almost all the state of the art models, we chose this dataset to proceed our work with. However, in order to adapt in our one-shot learning process, it must be partitioned. The training set and test set classes or categories in our suggested model cannot be repeated. As previously stated, the data collection is re-divided to meet our requirements.

4.3 Feature analysis

The extraction of the MFCC feature is a crucial stage in our one-shot learning process. There are many other features, such as standard deviation, variance, mean absolute, entropy, central frequency, mel frequency, number of peaks, and so on, however, not all are particularly used for speech handling. With all our research we have come across two features that work for speaker recognition tasks, one is mel spectrogram and other one is MFCCs. The objective of the mel spectrogram and MFCC is to transform a time domain signal into a frequency domain signal thus that we can grasp all of the data contained in voice sounds. However, just translating time domain signals to frequency domain signals isn't always the best solution. Mel-frequency spectrograms are similar to normal spectrograms (successive measurements recorded in short time frames), except they use Mel-frequency spacing. MFCC, on the other hand, are distinguished by the organization of the band pass in the mel frequency scale. Cepstrum outperforms the mel spectrogram because of its unique calculation. MFCCs need us to compute the Cepstrum, or the FFT of the log-magnitude or power spectrum, which is a "spectrum of a spectrum." Filtering in the time domain is similar to frequency domain multiplication, which is then added in the cepstrum. This enables better separation and investigation of the source and transmission channel. To put it another way, it functions like our ear when it comes to human comprehension from a machine. Our ear has a cochlea, which contains a lot of low-frequency filters and very few high-frequency filters. Mel filters can be used to imitate this. When we use this scale, our characteristics become more similar to what people hear. The goal of MFCC is to transform time domain signals into frequency domain signals by employing Mel filters to imitate the cochlea function. This frequency warping improves the audio representation. Also, MFCCs are created without much pre-processing from the raw audio input. This is the best feature where we came to our judgment based on our analysis.
We plotted a T-SNE graph figure 4.3 after we completed feature extraction to assess how strong our clustering was. It provided us a favorable result, as predicted, and then we moved on to the next phase.

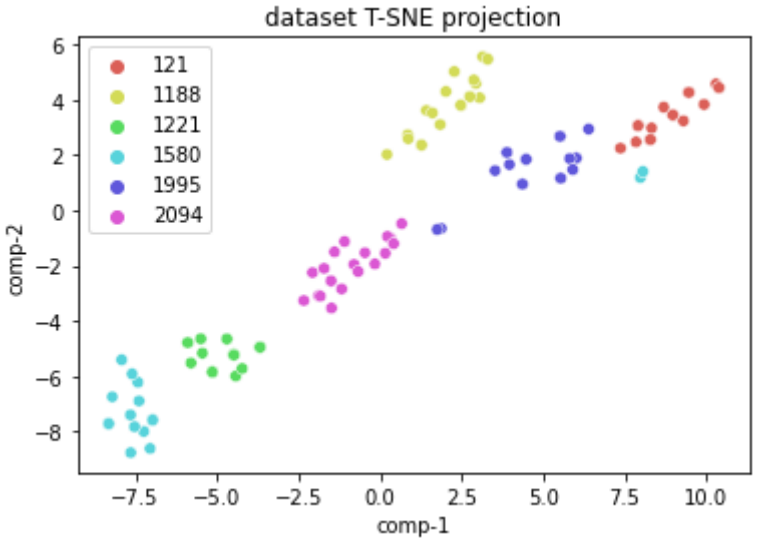


Figure 4.3: T-SNE plot for MFCCs feature

4.4 Algorithm Analysis

Most machine learning systems have made significant progress in recognizing people by their voices. However, to get similar performance, other strategies necessitate numerous cases. To address this issue, meta learning has led to a significant improvement in the categorization task. To categorize categories using a limited number of training instances, approaches such as few-shot learning, one-shot learning, and others are utilized. One-shot learning aims to classify fresh data after only hearing it once. This is useful when finding training instances is difficult. N-way one-shot categorization is commonly used in one-shot learning research. With one sample of each class, we sought to discern between them. In these instances, we can not train a classifier using traditional methods . Any sophisticated classification algorithm will rely on a much larger number of factors and function poorly on it.
In the classic learning framework, the model is trained from training data and then evaluated by classifying the testing data, however in our suggested methodology, the model will learn how to classify a given collection of training data and then evaluate using a set of testing data. The major reason we believe our model is superior to traditional approaches is that traditional models are unable to recognize data (classes) that have never been trained. It must train several samples of the same class to detect another sample from that identical class in order to improve accuracy. However, if we test samples from classes that have never been taught previously or have only been trained with one sample, the accuracy will be very low, as will be described and compared in the result analysis section. However, our model excels in this area since it can recognize a class that has never been observed or trained before.

The one-shot learning method employs various strategies to exploit different kinds of prior knowledge. Three of them, in particular, have grabbed our interest at the outset of our research: Prior knowledge of data, prior knowledge of similarity, and prior knowledge of learning. With the best analysis, we chose to work with prior

knowledge of similarity, which aids in the learning of embeddings in training tasks that tend to distinguish various classes even when they are not seen before. To find the similarity, the cosine similarity metric is used by the siamese network. So, after careful consideration, we have determined that the siamese network with triplet loss is the best fit for our model.

Chapter 5

Experimentation

The process below in figure 5.1 illustrates the step-by-step approach of the research that was conducted:

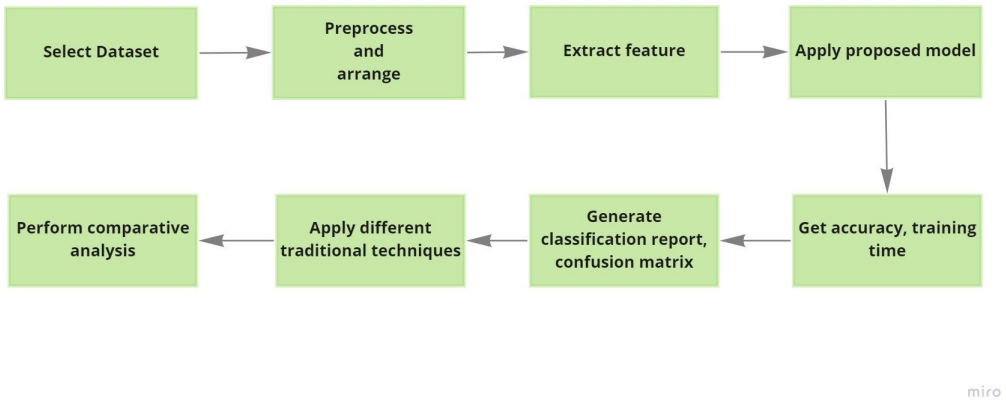


Figure 5.1: Workflow for the Experiment

Google Colab has been utilized to run all of the models. Colaboratory is a Google research project dedicated to the advancement of artificial intelligence research. The LibriSpeech ASR corpus was utilized. There are 12 subsets of voice samples, text files, and other data in the collection. However, the first job we had to complete was to customize our dataset to meet our requirements. Because the speciality of our technique is small datasets, we simply extracted features from three subsets of speech samples, which we separated into anchor, positive and negative numpy arrays (train) and support set and query set numpy arrays (test). After that, we put our model into action. The dense layer had size 80 and 40, and we utilized the activation function "relu in figure 5.2."

```
Model: "Embedding"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_11 (InputLayer)        [(None, 40)]              0
_________________________________________________________________
dense_31 (Dense)             (None, 80)                3280
_________________________________________________________________
batch_normalization_21 (Batc (None, 80)                320
_________________________________________________________________
dense_32 (Dense)             (None, 40)                3240
_________________________________________________________________
batch_normalization_22 (Batc (None, 40)                160
_________________________________________________________________
dense_33 (Dense)             (None, 40)                1640
=================================================================
Total params: 8,640
Trainable params: 8,400
Non-trainable params: 240
_________________________________________________________________
None
```

Figure 5.2: Dense layers in our model

The Adam optimizer was used with a value of 0.001. We completed 50 epochs in all, with a final loss of 2.4061 shown in figure 5.3. According to our study of training data, positive similarity was 0.984, or 98.4%, while negative similarity was 0.950, or 95%.

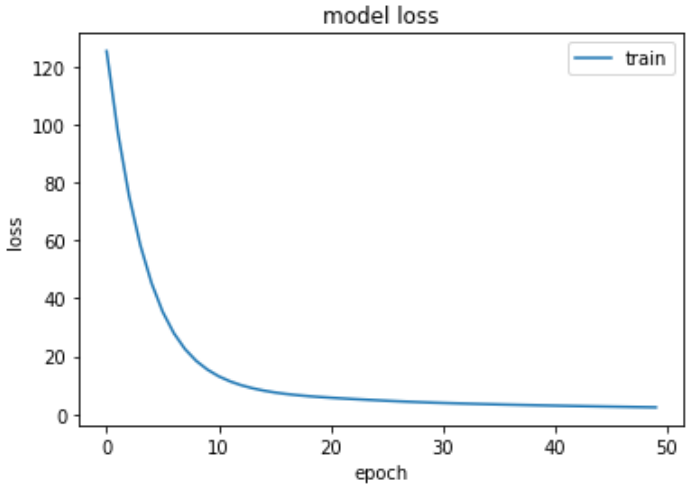


Figure 5.3: graph of loss with each epoch

Our model's training time was 6.43 seconds, which is extremely good for real-time application. Because our testing technique differs from the usual way, we use our model to evaluate fully untrained classes. We used several test cases to ensure accuracy in our findings. We started with two support sets (untrained class) each having only one sample. In addition, for each of those two classes, we had 25 samples

for a query set (untrained class). We get an overall accuracy of 1.0 or 100% after using cosine similarity in the testing phase. Then we expanded our support set to three untrained classes, each with one sample. In addition, there were three untrained classes in the query set, each with 25 samples. After testing it we got an accuracy of 0.95 or 95% . After that, we did a one-sample test for each of the four untrained classes. There were also four untrained classes in the query set, each with 25 samples. We got an accuracy of 0.84 or 84%, after testing it. Finally, we expanded our support set to five classes (untrained) with one sample each. There were also five untrained classes in the query set, each with 25 samples. We got an accuracy of 0.74, or 74%, after testing it. Following that, we compared our model to traditional techniques. K-Nearest Neighbor classifier, Random Forest classifier, and Multilayer Perceptron classifier are three of them. To do so, we combined three different training sets (anchor, positive, and negative) into a single dataset. In a conventional method, we would split this dataset into a training and testing set and train all of the classes to get near-perfect accuracy. To compare it to our model, we must see if these models can correctly predict samples that have never been trained, just as our model does without the classes being trained. However, as we all know, a conventional model can never correctly evaluate samples without its class being trained beforehand. So, in order to assess it more accurately, we gave these models the benefit of the doubt and had the classes (one sample each) trained beforehand, something we never do in our model. We did this by appending the support classes from our model to the dataset (one sample each) and training it with a 75:25 ratio. As expected, this gave us a very low accuracy compared to our model.

Chapter 6

Result Analysis

This paper offers a siamese network-based classification model for one-shot learning. However, for comparative analysis, different machine learning and neural network models were utilized, including Random Forest Classifier, Multilayer Perceptron Network, and K Nearest Neighbour Classifier. In addition, the assessment measures Precision, Recall, and F1-score were used in this study. The classification Report, which is a performance assessment statistic, shows the accuracy, precision, recall, and f1-score of the test data.

Accuracy: Accuracy is defined as the proportion of properly predicted observations to total observations. It is an effective classification measure when the datasets are symmetric or the levels of false positive and false negative are roughly identical (equation no 6.1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.1}$$

Precision: Precision is defined as the proportion of true positives to the total number of true and false positives. Precision (equation no 6.2) refers to a model's accuracy in terms of just how many anticipated positive values are really positive.

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

Recall: True positives to the total of true positives and false negatives is the definition of recall. Recall is a significant statistic when there is a cost involved with false negative data (equation no 6.3).

$$Recall = \frac{TP}{TP + FP} \tag{6.3}$$

F1-score: The weighted harmonic mean of precision and recall is used to get the f1-score. If the f1 in equation no 6.4 score is close to 1.00, the model's anticipated performance is superior. The F1-score achieves a good combination of accuracy and recall. When there is an unbalanced class allocation or a significant number of True Negative values, it is a helpful measure.

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{6.4}$$

6.1 Proposed Model:

As we used several test cases, we have several accuracy metrics in our findings.

For 2 classes in Support set:
In our proposed model, we get an overall accuracy of 1.0 or 100% for 2 classes after using cosine similarity in the testing phase.

Confusion matrix for 2 classes:

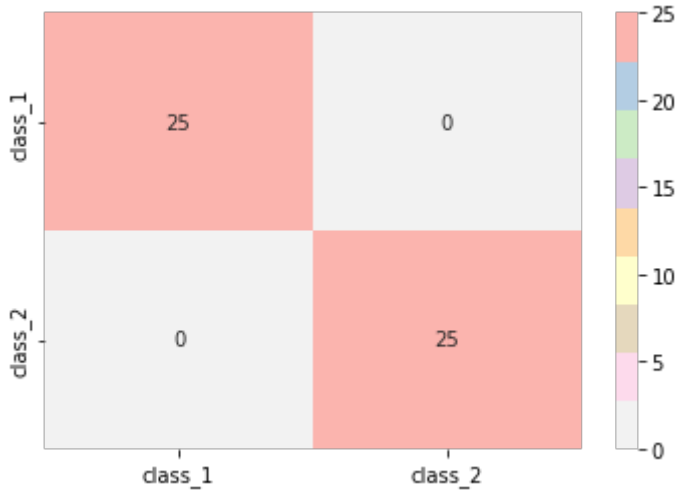


Figure 6.1: Confusion matrix for 2 classes

Here in figure 6.1 the confusion matrix, it is shown that class 1(speaker 1) and class 2(speaker 2) both have an accuracy of 1.00 or 100%.
The classification report for these 2 classes as follows:

	precision	recall	f1-score	support
237	1.00	1.00	1.00	25
61	1.00	1.00	1.00	25
accuracy			1.00	50
macro avg	1.00	1.00	1.00	50
weighted avg	1.00	1.00	1.00	50

Figure 6.2: Classification report for 2 classes

According to the figure 6.2 above, 237 and 61, 2 classes are represented. The precision, recall, f1-score are all 100%. The accuracy, macro average and weighted average is also 1.00 or 100%. It shows that our model perfectly works on such cases when there are 2 unknown classes.

For 3 classes in Support set:

After testing it with 3 untrained classes, we got an accuracy of 0.95 or 95% . The classification report for these 3 classes as follows:

Confusion matrix for 3 classes:

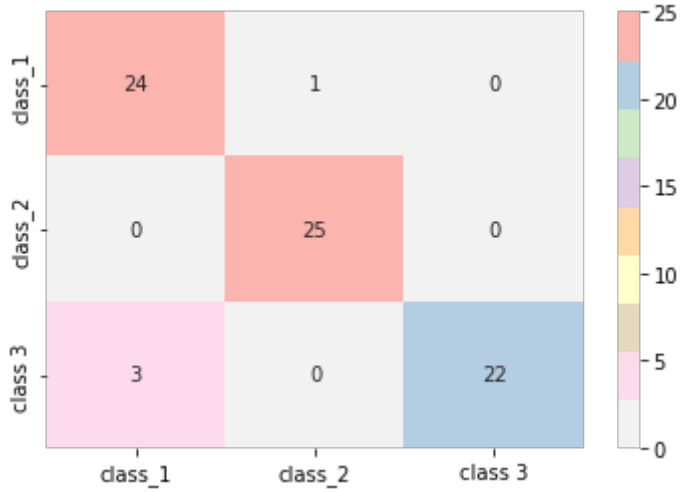


Figure 6.3: Confusion matrix for 3 classes

Here in figure 6.3 the confusion matrix, it is shown that for class 2(speaker 2) the accuracy is 1.00 or 100% as it classifies all the query samples perfectly. 1 sample is missed out in class 1(speaker 1) leading the accuracy to be 0.96 or 96%. And for class 3 (speaker 3) 3 samples are classified wrong leading the accuracy to be 0.88 or 88%.

The classification report for these 3 classes as follows:

	precision	recall	f1-score	support
1284	0.89	0.96	0.92	25
237	0.96	1.00	0.98	25
61	1.00	0.88	0.94	25
accuracy			0.95	75
macro avg	0.95	0.95	0.95	75
weighted avg	0.95	0.95	0.95	75

Figure 6.4: Classification report for 3 classes

According to the figure 6.4 above 1284, 237 and 61 respectively represent class 1(speaker 1), class 2(speaker 2) and class 3(speaker 3).The precision value is highest for class 3. Class 2 has the maximum recall and f1-score. All of the classes have relatively higher precision, recall and f1-score. The overall macro average of precision,

recall and f1-score are 0.95 or 95%.

For 4 classes in Support set: Then we did a one-sample test for each of the four untrained classes. We got an accuracy of 0.84 or 84%, after testing it.

Confusion matrix for 4 classes:

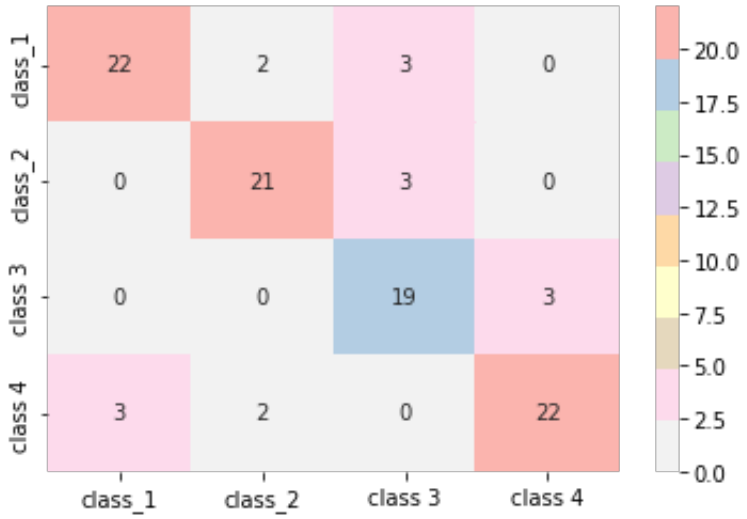


Figure 6.5: Confusion matrix for 4 classes

Here in figure 6.5 the confusion matrix, it is shown that 3 samples are missed out in class 1(speaker 1) and class 4(speaker 4) leading the accuracy to be 0.88 or 88%. For class 2, 4 samples are incorrectly classified and the accuracy here is 0.84 or 84%. For class 3 (speaker 3) 6 samples are classified wrong leading the accuracy to be 0.76 or 76%.
The classification report for these 4 classes as follows:

	precision	recall	f1-score	support
1089	0.81	0.88	0.85	25
1221	0.88	0.84	0.86	25
1580	0.86	0.76	0.81	25
2094	0.81	0.88	0.85	25
accuracy			0.84	100
macro avg	0.84	0.84	0.84	100
weighted avg	0.84	0.84	0.84	100

Figure 6.6: Classification report for 4 classes

According to the figure 6.6 above 1089,1221,1580 and 2094 respectively represent 38class 1(speaker 1), class 2(speaker 2), class 3(speaker 3) and class 4(speaker 4). The precision and f1-score are highest for class 2. The recall is highest for class 1 and class 4. All of the classes except have relatively higher precision, recall and

f1-score. The overall macro average of precision, recall and f1-score are 0.84 or 84%.

For 5 classes in Support set:

Finally, we expanded our support set to five classes (untrained) with one sample each. We got an accuracy of 0.74, or 74%, after testing it. Which in our model, is a bit low but we have future work introduced for this to get better accuracy.
Confusion matrix for 5 classes:

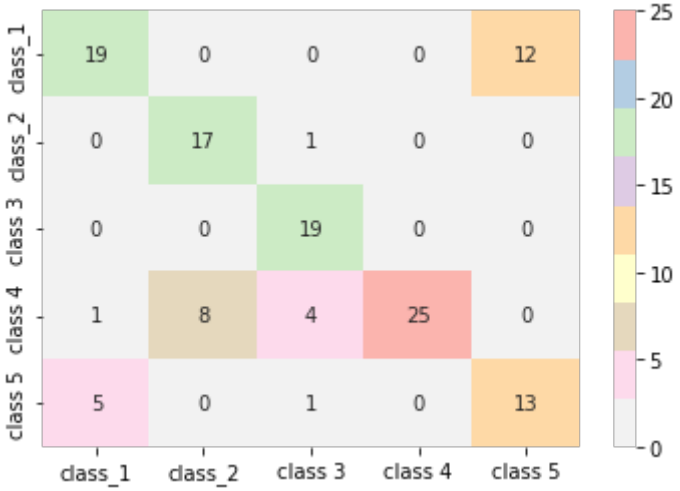


Figure 6.7: Confusion matrix for 5 classes

Here in figure 6.7 the confusion matrix, it is shown that for class 4(speaker 4) the accuracy is 1.00 or 100% as it classifies all the query samples perfectly. 6 samples are missed out in class 1(speaker 1) and class 3(speaker 3) leading the accuracy to be 0.76 or 76%. For class 2 (speaker 2) 8 samples are classified wrong leading the accuracy to be 0.68 or 68% and lastly for class 5(speaker 5) 12 classes are incorrectly classified leading to have the lowest accuracy of 0.52 or 52%.
The classification report for these 5 classes as follows:

	precision	recall	f1-score	support
1188	0.61	0.76	0.68	25
1284	0.94	0.68	0.79	25
1580	1.00	0.76	0.86	25
237	0.66	1.00	0.79	25
61	0.68	0.52	0.59	25
accuracy			0.74	125
macro avg	0.78	0.74	0.74	125
weighted avg	0.78	0.74	0.74	125

Figure 6.8: Classification report for 5 classes

According to the figure 6.8 above 1188, 1284,1580, 237 and 61 respectively represent class 1(speaker 1), class 2(speaker 2), class 3(speaker 3), class 4(speaker 4) and class

5(speaker 5). The precision value is highest for class 3. The recall value is highest for class 4 and f1-score is highest for class3. All of the classes have a mixture of highest/lowest precision, recall and f1-score. The overall macro average of precision is 0.78 or 78%. Overall macro average for recall and f1-score are 0.74 or 74%.

Combined Testing curve of proposed model in figure 6.9:

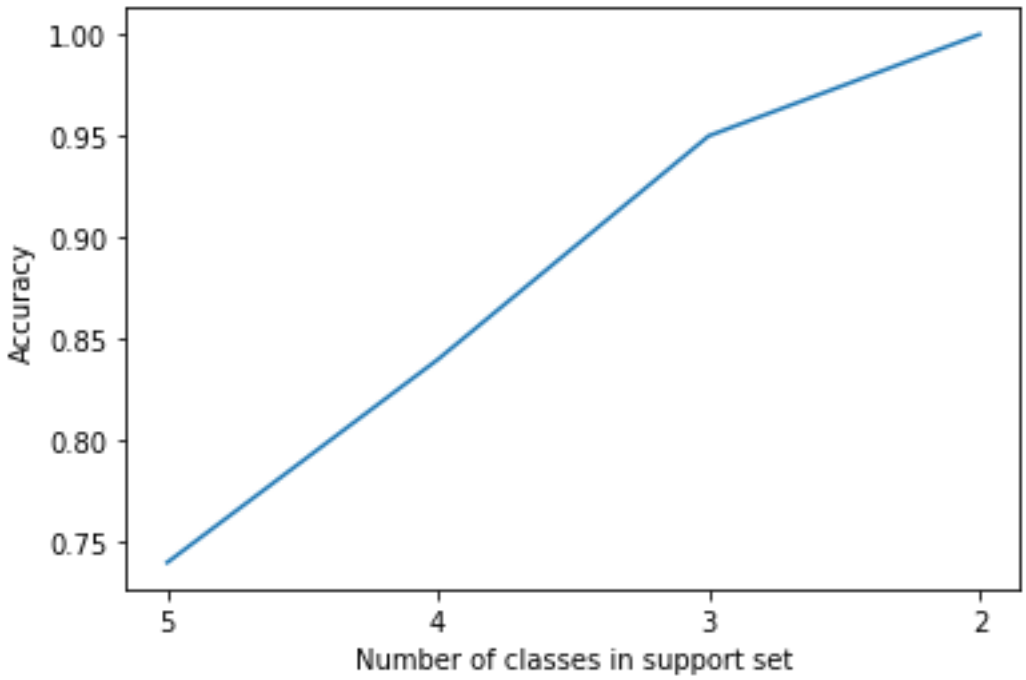


Figure 6.9: Accuracy plot for all support sets

The below table 6.1 shows overall accuracy for each class and time taken for training:

Support class	Query samples	Accuracy	Training Time(s)
2	50	1.00	6.43
3	75	0.95	
4	100	0.84	
5	125	0.74	

Table 6.1: Table for overall accuracy of proposed model

6.2 Combined analysis for other models

The aggregate categorization report for all models implemented is shown in the table 6.2 below.
It is made up of accuracy for all classes. We may infer from this table that the proposed model, i.e. one shot learning with siamese network, produces the greatest scores among the others.

Classifiers	2 classes	3 classes	4 classes	5 classes	(avg)Training time(s)
Proposed Model	1	0.95	0.84	0.74	6.43
KNN	0.76	0.73	0.71	0.67	13.61
Random Forest	0.29	0.29	0.29	0.3	11.92
MLP	0.22	0.14	0.44	0.2	69.41

Table 6.2: Combined analysis for all the models

When we compare accuracy for two classes thatis shown in figure 6.10, we obtain 0.76 or 76% for the KNN classifier, 0.29 or 29% for the Random Forest classifier, and 0.22 or 22% for the MLP classifier. Our model, on the other hand, provides exact accuracy of 1.0 or 100 %. When we compare accuracy for three classes, we get 0.73 (73%), 0.29 (29%), and 0.14 (14%), respectively, for the KNN, Random Forest, and MLP classifiers. Our model, on the other hand, has a 0.95 or 95 percent accuracy. When the KNN, Random Forest, and MLP classifiers are compared for four classes, we get 0.71 (71%), 0.29 (29%), and 0.44 (44%), respectively. Our model, on the other hand, has an accuracy of 0.84, or 84%. Lastly, we compared it for five classes and obtained an accuracy of 0.67 or 67% for KNN classifier, 0.3 or 30% for Random Forest classifier and 0.2 or 20% for MLP classifier. Here our model had an accuracy of 0.74 or 74%. We also compared the training time for each of the models and on an average got 6.43 seconds for our model, 13.61 seconds for KNN classifier, 11.92 seconds for Random Forest classifier and 69.41 seconds for MLP classifier.
Below is the accuracy graph for all the models:

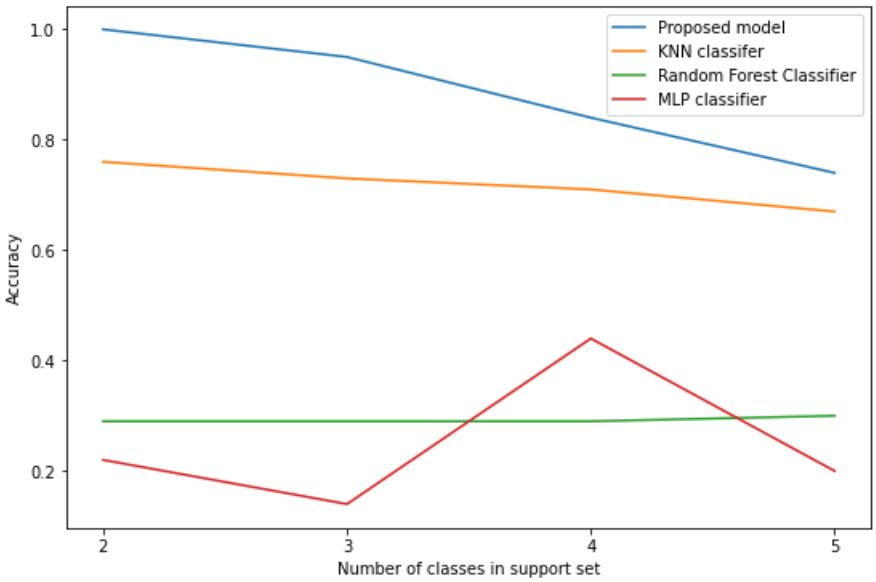


Figure 6.10: Graph for overall accuracy of all models

Chapter 7

7.1 Future Work

In the future work, we will assess the model's performance with a few additional modifications and tunings for larger numbers of classes. A contractive auto-encoder may also be used to extend the siamese network by learning a generalized embedding. The embeddings computed are utilized to train a prototype embedding using prototypical loss. This effort, we hope, will advance the implementation of the Siamese network. We'll also test it out on an embedded system to see how it performs in real-world scenarios.

7.2 Conclusion

In this paper, we explore the challenges of speaker recognition with limited training samples. We developed a one-shot learning strategy based on triplet loss to deal with the data shortage. As our supplied analysis and research statement demonstrate, the speaker recognition task can be completed with extremely few resources. This methodology will be useful in a variety of security situations when obtaining several samples is difficult. Although, based on our research, we believe that this sector may be utilized to verify new people who have never been seen previously. Self-training or our suggested model "one-shot learning" exhibit a large performance gain, as comprehensive experiments on a real-world dataset indicate minor variations between classes or domains of each speech.

Bibliography

[1] S. S. Stevens and J. Volkmann, "Newman. eb a scale for the measurement of the psychological magnitude pitch," *Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[2] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the IRE*, vol. 49, no. 1, pp. 8–30, 1961.

[3] P. D. Bricker and S. Pruzansky, "Effects of stimulus content and duration on talker identification," *The Journal of the Acoustical Society of America*, vol. 40, no. 6, pp. 1441–1449, 1966.

[4] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[5] P. Papamichalis and G. Doddington, "A speaker recognizability test," in *ICASSP'84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 9, 1984, pp. 93–96.

[6] A. Schmidt-Nielsen and K. R. Stern, "Identification of known voices as a function of familiarity and narrow-band coding," *The Journal of the Acoustical Society of America*, vol. 77, no. 2, pp. 658–663, 1985.

[7] D. Van Lancker, J. Kreiman, and K. Emmorey, "Familiar voice recognition: Patterns and parameters part i: Recognition of backward voices," *Journal of phonetics*, vol. 13, no. 1, pp. 19–38, 1985.

[8] D. Van Lancker and J. Kreiman, "Voice discrimination and recognition are separate abilities," *Neuropsychologia*, vol. 25, no. 5, pp. 829–834, 1987.

[9] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[10] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[11] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods," 2000.

[14] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[15] A. Schmidt-Nielsen and T. H. Crystal, "Speaker verification by human listeners: Experiments comparing human and machine performance using the nist 1998 speaker evaluation data," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 249–266, 2000.

[16] F. A. Gers and E. Schmidhuber, "Lstm recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[17] M. Brown, D. G. Lowe, *et al.*, "Recognising panoramas.," in *ICCV*, vol. 3, 2003, p. 1218.

[18] T. Kinnunen, "Spectral features for automatic text-independent speaker recognition," *Licentiate's thesis*, 2003.

[19] Z. Wu and Z. Cao, "Improved mfcc-based feature for robust speaker identification," *Tsinghua Science & Technology*, vol. 10, no. 2, pp. 158–161, 2005.

[20] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[22] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[23] J. Gudnason and M. Brookes, "Voice source cepstrum coefficients for speaker identification," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2008, pp. 4821–4824.

[24] I. Shahin, "Speaker identification in the shouted environment using suprasegmental hidden markov models," *Signal Processing*, vol. 88, no. 11, pp. 2700–2708, 2008.

[25] M. Alsulaiman, G. Muhammad, Y. Alotaibi, A. Mahmood, and M. A. Bencherif, "Building a speaker recognition with one sample," in *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT'09) Huangshan, People's Republic of China*, 2009, pp. 26–28.

[26] F. Eyben, M. Wöllmer, B. Schuller, and A. Graves, "From speech to letters-using a novel neural network architecture for grapheme based asr," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, IEEE, 2009, pp. 376–380.

[27] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information processing systems*, vol. 22, pp. 1096–1104, 2009.

[28] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, Makuhari, vol. 2, 2010, pp. 1045–1048.

[29] L. Wang, K. Minami, K. Yamamoto, and S. Nakagawa, "Speaker identification by combining mfcc and phase information in noisy environments," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2010, pp. 4502–4505.

[30] K. Youssef, S. Argentieri, and J.-L. Zarader, "Binaural speaker recognition for humanoid robots," in *2010 11th International Conference on Control Automation Robotics & Vision*, IEEE, 2010, pp. 2295–2300.

[31] K. Daqrouq, "Wavelet entropy and neural network for text-independent speaker identification," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 796–802, 2011.

[32] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011. DOI: 10.1109/TASL.2010.2064307.

[33] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, 2011.

[34] A. Lawson, P. Vabishchevich, M. Huggins, P. Ardis, B. Battles, and A. Stauffer, "Survey and evaluation of acoustic features for speaker recognition," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011, pp. 5444–5447.

[35] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "A multi-stream asr framework for blstm modeling of conversational speech," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011, pp. 4860–4863.

[36] F. Grondin and F. Michaud, "Wiss, a speaker identification system for mobile robots," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 1817–1822.

[37] B. Lake, R. Salakhutdinov, and J. Tenenbaum, "Concept learning as motor program induction: A large-scale empirical study," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 34, 2012.

[38] N. Singh, P. R. Khan, and R. S. Pandey, "Applications of speaker recognition," *Procedia Engineering*, vol. 38, pp. 3122–3126, Dec. 2012. DOI: 10.1016/j.proeng.2012.06.363.

[39] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*, IEEE, 2013, pp. 273–278.

[40] M. Srikanth, "Speaker verification and keyword spotting systems for forensic applications," Ph.D. dissertation, INDIAN INSTITUTE OF TECHNOLOGY MADRAS, 2013.

[41] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *2014 22nd European signal processing conference (EUSIPCO)*, IEEE, 2014, pp. 506–510.

[42] G. Nijhawan and M. Soni, "Speaker recognition using mfcc and vector quantisation," *International Journal on Recent Trends in Engineering & Technology*, vol. 11, no. 2, p. 211, 2014.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[44] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2014, pp. 4052–4056.

[45] K. Daqrouq and T. A. Tutunji, "Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers," *Applied Soft Computing*, vol. 27, pp. 231–239, 2015.

[46] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[47] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, vol. 2, 2015.

[48] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[49] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, Citeseer, vol. 8, 2015, pp. 18–25.

[50] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*, IEEE, 2015, pp. 1–6.

[51] T. Sainath, R. J. Weiss, K. Wilson, A. W. Senior, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," 2015.

[52] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, PMLR, 2016, pp. 173–182.

[53] G. Gelly, J.-L. Gauvain, V. B. Le, and A. Messaoudi, "A divide-and-conquer approach for language identification based on recurrent neural networks.," in *Interspeech*, San Francisco, CA, 2016, pp. 3231–3235.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[55] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5115–5119.

[56] Y. Lukic, C. Vogt, O. Dürr, and T. Stadelmann, "Speaker identification and clustering using convolutional neural networks," in *2016 IEEE 26th international workshop on machine learning for signal processing (MLSP)*, IEEE, 2016, pp. 1–6.

[57] C. Raffel and D. P. Ellis, "Pruning subsequence search with attention-based embedding," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 554–558.

[58] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2016, pp. 165–170.

[59] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[60] G. Bhattacharya, M. J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification.," in *Interspeech*, 2017, pp. 1517–1521.

[61] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: An end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[62] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[63] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017, pp. 999–1003.

[64] Y. Wang, X. Deng, S. Pu, and Z. Huang, "Residual convolutional ctc networks for automatic speech recognition," *arXiv preprint arXiv:1702.07793*, 2017.

[65] J. Yuan, H. Guo, Z. Jin, H. Jin, X. Zhang, and J. Luo, "One-shot learning for fine-grained relation extraction via convolutional siamese neural network," in *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, 2017, pp. 2194–2199.

[66] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances.," in *Interspeech*, 2017, pp. 1487–1491.

[67] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv preprint arXiv:1701.02720*, 2017.

[68] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Stock market forecast using multivariate analysis with bidirectional and stacked (lstm, gru)," in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, IEEE, 2018, pp. 1–7.

[69] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[70] A.-L. Georgescu and H. Cucu, "Gmm-ubm modeling for speaker recognition on a romanian large speech corpora," in *2018 International Conference on Communications (COMM)*, IEEE, 2018, pp. 547–551.

[71] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843.

[72] P. Manocha, R. Badlani, A. Kumar, A. Shah, B. Elizalde, and B. Raj, "Content-based representations of audio using siamese neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 3136–3140.

[73] A. Mobiny and M. Najarian, "Text-independent speaker verification using long short-term memory networks," *arXiv preprint arXiv:1805.00604*, 2018.

[74] H. Muckenhirn, M. M. Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using cnns," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4884–4888.

[75] N. Singh, A. Agrawal, and P. R. Khan, "Principle and applications of speaker recognition security system," Jun. 2018.

[76] I. Vélez, C. Rascon, and G. Fuentes-Pineda, "One-shot speaker identification for a service robot using a cnn-based generic verifier," *arXiv preprint arXiv:1809.04115*, 2018.

[77] C.-i. Wang and G. Tzanetakis, "Singing style investigation by residual siamese convolutional neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 116–120.

[78] Y. Zhang, B. Pardo, and Z. Duan, "Siamese style convolutional neural networks for sound search by vocal imitation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 429–441, 2018.

[79] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, 2019.

[80] R. Eloff, H. A. Engelbrecht, and H. Kamper, "Multimodal one-shot learning of speech and images," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 8623–8627.

[81] R. Jagiasi, S. Ghosalkar, P. Kulal, and A. Bharambe, "Cnn based speaker recognition in language and text-independent small scale system," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, IEEE, 2019, pp. 176–179.

[82] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *Advances in Computers*, 1, vol. 117, Elsevier, 2020, pp. 339–368.

[83] S. Kim, I. Kim, L. F. Vecchietti, and D. Har, "Pose estimation utilizing a gated recurrent unit network for visual localization," *Applied Sciences*, vol. 10, no. 24, p. 8876, 2020.

[84] D.-R. Liu, S.-J. Lee, Y. Huang, and C.-J. Chiu, "Air pollution forecasting based on attention-based lstm neural network and ensemble learning," *Expert Systems*, vol. 37, no. 3, e12511, 2020.

[85] M. A. Khan, "Hcrnnids: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, p. 834, 2021.

[86] A. Shafik, A. Sedik, B. Abd El-Rahiem, E.-S. M. El-Rabaie, G. M. El Banby, F. E. Abd El-Samie, A. A. Khalaf, O.-Y. Song, and A. M. Iliyasu, "Speaker identification based on radon transform and cnns in the presence of different types of interference for robotic applications," *Applied Acoustics*, vol. 177, p. 107 665, 2021.

[87] X. Zhang, H. Kuehnelt, and W. De Roeck, "Traffic noise prediction applying multivariate bi-directional recurrent neural network," *Applied Sciences*, vol. 11, no. 6, p. 2714, 2021.

[88] R. O. Ogundokun, O. C. Abikoye, A. A. Adegun, and J. B. Awotunde, "Speech recognition system: Overview of the state-of-the-arts,"